

**Ч'НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

«На правах рукопису»  
УДК 004.932

«До захисту допущено»  
Науковий керівник кафедри  
\_\_\_\_\_ І.А. Дичка  
«\_\_» \_\_\_\_\_ 2018 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**зі спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Спосіб та програмне забезпечення для комплексної оцінки  
та покращення якості зображень з використанням нейронних мереж»**

Виконав:

студент VI курсу, групи КП-71мп  
Лампіга Юрій Анатолійович \_\_\_\_\_

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,  
Сулема Є. С. \_\_\_\_\_

Рецензент:

Доцент кафедри АСОІУ, к.т.н., доцент,  
Муха І.П. \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

Київ – 2018 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою  
Спеціальність (спеціалізація) – 121 «Інженерія програмного забезпечення»  
(«Програмне забезпечення комп'ютерних та інформаційно-пошукових систем»)

ЗАТВЕРДЖУЮ

Науковий керівник  
кафедри

\_\_\_\_\_ І.А. Дичка

«\_\_» \_\_\_\_\_ 2018  
р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**

Лампізі Юрію Анатолійовичу

1. Тема дисертації «Спосіб та програмне забезпечення для комплексної оцінки та покращення якості зображень з використанням нейронних мереж», науковий керівник дисертації доцент кафедри ПЗКС, к.т.н., Сулема Є.С затверджені наказом по університету від «\_\_» \_\_\_\_\_ 2018 р. № \_\_\_\_\_
2. Термін подання студентом дисертації «14» грудня 2018 р.
3. Об'єкт дослідження: процес комплексної оцінки та покращення якості зображень з використанням нейронних мереж.
4. Предмет дослідження: способи та засоби комплексної оцінки та покращення якості зображень з використанням нейронних мереж.
5. Перелік завдань, які потрібно виконати:
  - провести аналіз існуючих метрик якості зображень;
  - провести аналіз алгоритмів, способів та засобів покращення якості зображень;
  - запропонувати новий спосіб комплексної оцінки та покращення якості зображень з використанням нейронних мереж;
  - розробити програмне забезпечення, що реалізує запропонований спосіб;
  - провести дослідження ефективності розробленого програмного забезпечення.
6. Орієнтовний перелік публікацій:
  - Тези доповіді на конференції ПМК-2018 «Покращення якості зображень з використанням нейронних мереж».

7. Дата видачі завдання «04» жовтня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Грунтовне ознайомлення з предметною галуззю	17.12.2017	
2.	Визначення структури магістерської дисертації; аналіз предметної галузі, вивчення літератури	04.12.2018	
3.	Робота над першим розділом магістерської дисертації; постановка задачі, проведення наукового дослідження	17.02.2018	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	14.05.2018	
5.	Проведення наукового дослідження; робота над тезами доповіді за результатами наукового дослідження	01.10.2018	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації; підготовка матеріалів доповіді на конференції ПМК-2018.	20.10.2018	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка матеріалів доповіді на конференції ПМК-2018	05.11.2018	
8.	Оформлення текстової і графічної частини магістерської дисертації	06.12.2018	

Студент

Ю.А. Лампіга

Науковий керівник дисертації

Є.С. Сулема

## ABSTRACT

**Relevance.** The volume of digital information is constantly increasing. In addition, with the development of software and hardware technologies, the requirements for the quality of digital data, in particular the quality of images, are raising. At the same time, the popularity of data processing methods with the artificial intelligence algorithms and deep learning are growing, in particular with neural networks. A large number of neural network architectures for image processing have been proposed. The analysis of the proposed solutions shows that there is still no such model that would have best subjective and objective metrics. Therefore, the development of a method for comprehensive evaluation and image quality improvement with neural networks is relevant.

**The object of research** in this paper there is a process of complex image evaluation and image quality improvement using neural networks.

**The subject of the study** are methods and software for the complex image quality evaluation and image quality improvement using neural networks.

**The purpose of the study** is an analysis of existing methods for image quality evaluation and image quality improvement with neural networks, their properties, features, and the further implementation of the method.

**Research methods.** Statistical and empirical methods.

**The scientific novelty** of the work is as follows:

1. Proposed image quality improvement method uses convolutional neural networks that does not require training with large datasets, unlike the existing methods.
2. Proposed complex image quality evaluation method uses values of metrics PSNR, SSIM, MSE, unlike the existing methods.

**The practical value** of the work is to use the results of research for efficient usage of convolutional neural networks for the image processing.

**Test work.** The main provisions and results of work were reported and discussed at the Xth International Conference of Masters and Postgraduate Students "Applied Mathematics and Computer", PMK-2018.

**Structure and scope of work.** The master's dissertation consists of an introduction, five sections, conclusions and appendices.

The introduction provides a general description of the work, an assessment of the current state of the problem, the relevance of the research direction.

The first section contains the existing methods for image quality improvement analysis.

The second section contains the review of complex image analysis and image quality improvement with neural networks, and neural network architecture.

The third section contains the software architecture, the definite algorithm.

The fourth section contains the results of development and shows images processed by the developed method.

The fifth section presents the construction of a business model that justifies the feasibility of the software and predicts its potential profitability in the future.

The conclusions are analyzed the results of work.

The work is performed on 77 sheets and contains the list of used sources.

**Keywords:** neural network, image processing, computer graphics, convolutional network.

## РЕФЕРАТ

**Актуальність.** Обсяги цифрової інформації постійно зростають. Крім того, з розвитком програмних та апаратних технологій, підвищуються загальні вимоги до якості цифрових даних, зокрема до якості зображень. Водночас, поширення набувають методи обробки даних за допомогою алгоритмів штучного інтелекту та глибокого навчання, зокрема з використанням нейронних мереж. На сьогодні запропоновано велику кількість архітектур нейронних мереж для обробки зображень. Аналіз запропонованих рішень показує, що й досі не існує такої моделі, яка б була кращою за суб'єктивними та об'єктивними метрикам. Тому розроблення способу для комплексної оцінки та покращення якості зображень з використання нейронних мереж є актуальним.

**Об'єктом дослідження** є процес комплексної оцінки та покращення якості зображень з використанням нейронних мереж.

**Предметом дослідження** є способи та засоби комплексної оцінки та покращення якості зображень з використанням нейронних мереж.

**Метою** дослідження є аналіз існуючих способів комплексної оцінки та покращення якості зображень з використанням нейронних мереж, їх властивостей, особливостей з подальшим створенням вдосконаленого способу.

**Методи дослідження.** В роботі використовуються статистичні та емпіричні методи.

**Наукова новизна** роботи полягає в наступному:

1. Запропоновано спосіб покращення якості зображень, який на відміну від існуючих способів використовує згорткову нейронну мережу, яка не потребує навчання на великих наборах даних.

2. Запропоновано спосіб комплексної оцінки якості зображень, який на відміну від існуючих способів використовує усереднене значення метрик PSNR, SSIM, MSE.

**Практична цінність** роботи полягає у можливості застосування отриманих результатів для ефективного використання згорткової нейронної мережі для задачі обробки зображень.

**Апробація роботи.** Основні положення і результати роботи доповідалися та обговорювалися на X науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2018.

**Структура та обсяг роботи.** Магістерська дисертація складається з вступу, п'яти розділів, висновків та додатків.

У вступі наведена загальна характеристика роботи та наведено сферу застосування розробленого способу.

У першому розділі проведено аналіз існуючих методів, способів та алгоритмів, які стосуються аналізу та обробки зображень.

У другому розділі описано спосіб комплексної оцінки та покращення якості зображень з використанням нейронних мереж та архітектуру розробленої нейронної мережі.

У третьому розділі описано архітектуру програмного забезпечення, визначено кінцевий алгоритм роботи, описано формат вхідних даних, з якими буде працювати реалізоване програмне забезпечення.

Четвертий розділ містить результати роботи програмного засобу та результати його порівняння з існуючими засобами.

У п'ятому розділі наведено побудову бізнес-моделі, що обґрунтовує доцільність реалізованого програмного забезпечення та прогнозує його потенційну прибутковість у майбутньому.

У висновках проаналізовано отримані результати роботи.

Робота виконана на 77 аркушах, містить посилання на список використаних літературних джерел.

**Ключові слова:** нейронна мережа, комп'ютерна графіка, обробка зображень, згорткова нейронна мережа.



## РЕФЕРАТ

**Актуальность.** Объемы цифровой информации постоянно растут. Помимо этого, с развитием программных и аппаратных технологий, повышаются общие требования к качеству цифровых данных, в частности к качеству изображений. В то же время, распространение получают методы обработки данных с помощью алгоритмов искусственного интеллекта и глубокого обучения, в частности с использованием нейронных сетей. На сегодня предложено большое количество архитектур нейронных сетей для обработки изображений. Анализ предлагаемых решений показывает, что до сих пор не существует такой модели, которая была лучше по субъективным и объективным метрикам. Поэтому разработка способа для комплексной оценки и улучшения качества изображений с использованием нейронных сетей является актуальной.

**Объектом исследования** является процесс комплексной оценки и улучшения качества изображений с использованием нейронных сетей.

**Предметом исследования** являются способы и средства комплексной оценки и улучшения качества изображений с использованием нейронных сетей.

**Целью исследования** является анализ существующих способов комплексной оценки и улучшения качества изображений с использованием нейронных сетей, их свойств, особенностей с последующим созданием собственного способа.

**Методы исследования.** В работе используются статистические и эмпирические методы.

**Научная новизна** работы заключается в следующем:

1. Предложен способ улучшения качества изображений, который в отличие от существующих способов использует згортковых

нейронную сеть, которая не требует обучения на больших наборах данных.

2. Предложен способ комплексной оценки качества изображений, который в отличие от существующих способов использует усредненное значение метрик PSNR, SSIM, MSE.

**Практическая ценность** работы состоит в возможности применения полученных результатов для эффективного использования свёрточной нейронной сети для обработки изображений.

**Апробация работы.** Основные положения и результаты работы докладывались и обсуждались на X научной конференции магистрантов и аспирантов «Прикладная математика и компьютеринг» ПМК-2018.

**Структура и объем работы.** Магистерская диссертация состоит из введения, пяти разделов, заключения и приложений.

Во введении приведена общая характеристика работы.

В первом разделе проведен анализ существующих методов, способов и алгоритмов касающихся анализа и обработки изображений, как с использованием нейронных сетей, так и без них.

Во втором разделе описан способ комплексной оценки и улучшения качества изображений с помощью нейронных сетей, и архитектуру разработанной нейронной сети.

В третьем разделе описано архитектуру программного обеспечения, определены конечный алгоритм работы.

В четвертом разделе содержит результаты работы программного средства и проведено сравнение с существующими средствами.

В пятом разделе приведено построение бизнес-модели, обоснована целесообразность реализованного программного обеспечения, приведён прогноз её потенциальной прибыльности в будущем.

В выводах проанализированы полученные результаты работы.

Работа выполнена на 77 страницах, содержит список использованных литературных источников.

**Ключевые слова:** нейронная сеть, компьютерная графика, обработка изображений, свёрточная нейронная сеть.

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ, ПОЗНАЧЕНЬ.....	3
ВСТУП .....	4
1. ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ ОБРОБКИ ЗОБРАЖЕНЬ ..	6
1.1 Метод найближчого сусіда.....	7
1.2 Бікубічна інтерполяція .....	7
1.3 Super-Resolution Convolutional Neural Network.....	8
1.4 Rapid and Accurate Image Super-Resolution.....	12
1.5 Нейронні мережі.....	15
1.6 Висновок до першого розділу.....	20
2. КОМПЛЕКСНА ОЦІНКА ТА ПОКРАЩЕННЯ ЯКОСТІ ЗОБРАЖЕНЬ .....	21
2.1 Комплексна оцінка якості зображень .....	21
2.2 Архітектура нейронної мережі .....	24
2.3 Спосіб покращення якості зображень.....	29
2.4 Висновок до другого розділу .....	30
3. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СПОСОБУ .....	31
3.1 Необхідні засоби програмної реалізації .....	31
3.2 Модуль комплексної оцінки зображень .....	34
3.3 Розроблене програмне забезпечення .....	35
3.4 Висновок до третього розділу .....	37
4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ .....	38
4.1 Методика оцінювання .....	38
4.2 Дослідження для визначення кількості шарів нейронної мережі та аргументів мережі .....	38
4.3 Результати роботи програмного засобу.....	40
4.4 Візуальне порівняння з іншими способами .....	41
4.5 Порівняння результатів роботи за показником PSNR .....	41
4.6 Висновок до четвертого розділу.....	44

5.	РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ .....	45
5.1	Опис проблеми та дерево проблем. ....	45
5.2	Аналіз зацікавлених сторін проекту. ....	49
5.3	Опис наукового проекту та технології. ....	54
5.4	Бізнес рішення та основні характеристики бізнес - продукту. ....	55
5.5	Доходи та витрати.....	64
5.6	Бізнес-модель.....	71
5.7	Висновок до п'ятого розділу.....	73
	ВИСНОВОК.....	74
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ .....	75

## **СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ, ПОЗНАЧЕНЬ**

PSNR – пікове співвідношення сигналу до шуму (Peak Signal-to-Noise Ratio, PSNR) або співвідношення між максимумом можливого значення сигналу та потужністю шуму, що спотворює значення корисного сигналу.

MSE – (Mean squared error) середня квадратична похибка.

SSIM – (Structure similarity) – індекс структурної подібності.

ІНМ – штучна нейронна мережа.

CSV – (character-separated values) файловий формат який використовує кому або крапку з комою для того щоб розділяти дані у файлі.

CPU – (central processor unit) центральний процесор електронно обчислювальної машини.

RAM – швидкодіюча пам'ять, призначена для запису, зберігання та читання інформації у процесі її обробки.

IoT – Інтернет речей

## ВСТУП

З кожним роком зростає кількість і складність задач з оброблення зображень, що зумовлено потребою в автоматизації, необхідністю зменшення трафіку в мережі Інтернет, появою IoT пристроїв, які працюють із зображеннями. Тому розробка способу для комплексного аналізу та покращення якості зображень у комп'ютерних системах є актуальною задачею. Один з перспективних напрямків вирішення даної проблеми ґрунтується на застосуванні штучних нейронних мереж.

На сьогодні запропоновано велику кількість архітектур нейронних мереж, що застосовуються для розпізнавання та класифікації об'єктів, проте лише невелика кількість штучних нейронних мереж спрямована саме на вирішення задачі покращення якості зображень. Аналіз існуючих рішень показав, що й досі не існує способу або засобу, який був би кращим за кількісними показниками об'єктивних метрик якості зображень, або застосовним в портативних пристроях.

Останнім часом особлива увага приділяється глибоким згортковим нейронним мережам. Зокрема, вони використовуються для аналізу вмісту зображення, класифікації об'єктів на цифровому зображенні та обробки зображень. Глибокі згорткові мережі є ефективним інструментом для створення, обробки, відновлення зображень. Вони також застосовуються для аналізу зображень з метою розпізнавання та класифікації об'єктів, образів та тексту.

Переваги згорткових мереж над багатошаровими полягають у використанні спільної ваги у згорткових шарах, що означає, що для кожного пікселя шару використовується один і той же фільтр, що підвищує швидкодію згорткових мереж відносно багатошарових.

В межах даної роботи виконано дослідження, що включає комплексну оцінку растрових зображень, вибір та застосування метрик для оцінки якості оброблених зображень порівняно з оригіналом, а також оброблення зображень з метою покращення якості зображень.

В результаті проведеного дослідження запропоновано спосіб, що складається з комплексної оцінки зображення і покращення його якості засобами нейронної мережі та фільтрів. Під покращенням в контексті даної роботи розуміється підвищення роздільності вхідного зображення.



## 1. ПОРІВНЯЛЬНИЙ АНАЛІЗ СПОСОБІВ ОБРОБКИ ЗОБРАЖЕНЬ

Підвищення роздільності зображень, тобто створення фото високої роздільності на основі одного фото низької роздільності – відома наукова проблема. Вона важлива для багатьох задач: збільшення масштабу фото і тексту, проекції відео на великий екран, тощо. Також даний підхід дозволяє розпізнати номер автомобіля або обличчя на цифровому фото, за допомогою способів, які займаються відновленням або створення подібного до реальності зображення завдяки початковому фотознімку низької якості.

Згідно проведеного аналізу, підвищення роздільності зображень низької якості вважається однією з найбільш складних та важливих проблем серед задач з оброблення зображень – як з теоретичної, так і з практичної точки зору.

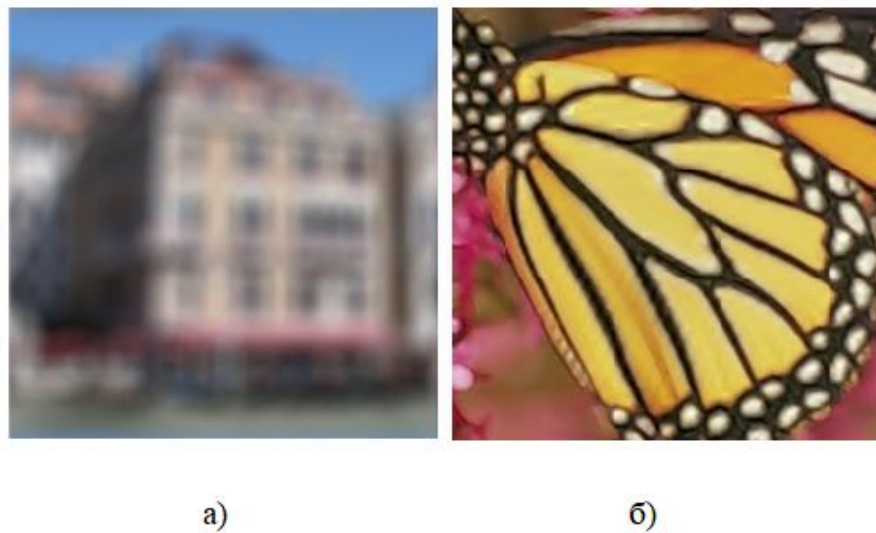


Рис 1.1 – а) зображення з не вірно налаштованим фокусом на фото камері; б) зображення низького розширення (256x256 пікселів, отримане з зображення 1024x1024 пікселів)

Окремими випадками є розмиття через невірно встановлену точку фокусу, низька роздільність зображення отримана в наслідок переведення аналогової інформації в цифрову або передачі інформацію по мережам, які можуть впливати на якісну характеристику інформації (рис. 1.1).

### 1.1 Метод найближчого сусіда

Метод найближчого сусіда класифікують як самий простий та швидкодіючий метод підвищення роздільності зображення. Метод передбачає наступні дії: для кожного пікселя вихідного зображення вибирається один піксель вхідного найближчий до положення вихідного, враховуючи масштабування [3]. Даний метод дає на виході надто пікселізоване зображення при збільшенні масштабу і дуже зернисте зображення при зменшенні масштабу.

В контексті обробки зображень візуальна складова методу найближчого сусіда для підбору кольорів при масштабуванні виглядає наступним чином рис. 1.2.

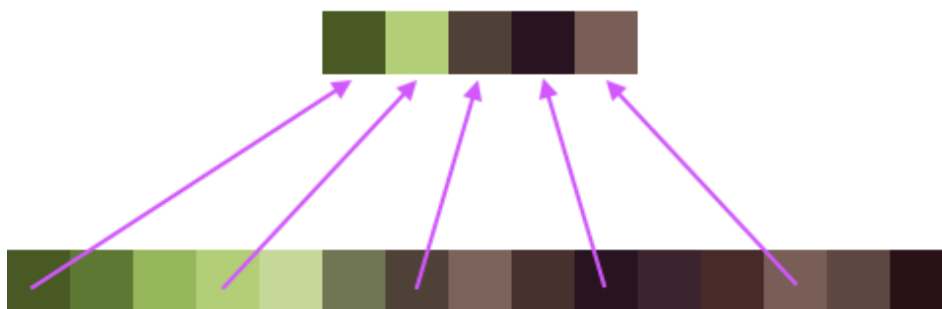


Рис 1.2– Підбір кольорів методом найближчого сусіду

### 1.2 Бікубічна інтерполяція

Поверхня, яка отримується в результаті бікубічної інтерполяції є гладкою функцією на межах сусідніх квадратів, на відміну від поверхонь,

отриманих в результаті білінійної інтерполяції або безпосередньо інтерполяції методом найближчого сусіда [4].

Перевага бікубічної інтерполяції над білінійною складається в тому, що бікубічна бере до уваги масив оточуючих пікселів, результат роботи бікубічної інтерполяції відображено в даній роботі у розділі 4.

Бікубічна інтерполяція часто використовується в обробці зображень. В результаті її застосування отримується об'єктивно більш якісне зображення в порівнянні зображенням, яке пройшло обробку білінійною інтерполяцією. Також бікубічна інтерполяція застосовується в алгоритмах управління верстатів з ЧПУ для обліку нерівностей площин, наприклад, при фрезеруванні друкованих плат [5].

### **1.3 Super-Resolution Convolutional Neural Network**

Даний метод підвищення роздільності зображень використовує штучні нейронні мережі, фільтри та навчання для обробки зображень.

Перед тим, як здійснювати збільшення роздільності зображення за допомогою даної моделі, зображення попередньо збільшується до необхідних розмірів. Як правило, при цьому використовується алгоритм бікубічної інтерполяції. Хоча формально даний метод інтерполяції і є згорткою, а самі автори дослідження [6] стверджують, що до нього може бути застосована формулювання «шар», в описі архітектури нейронної мережі операція згортка звичайно не фігурує. Архітектура мережі SRCNN відображено на рис. 1.3.

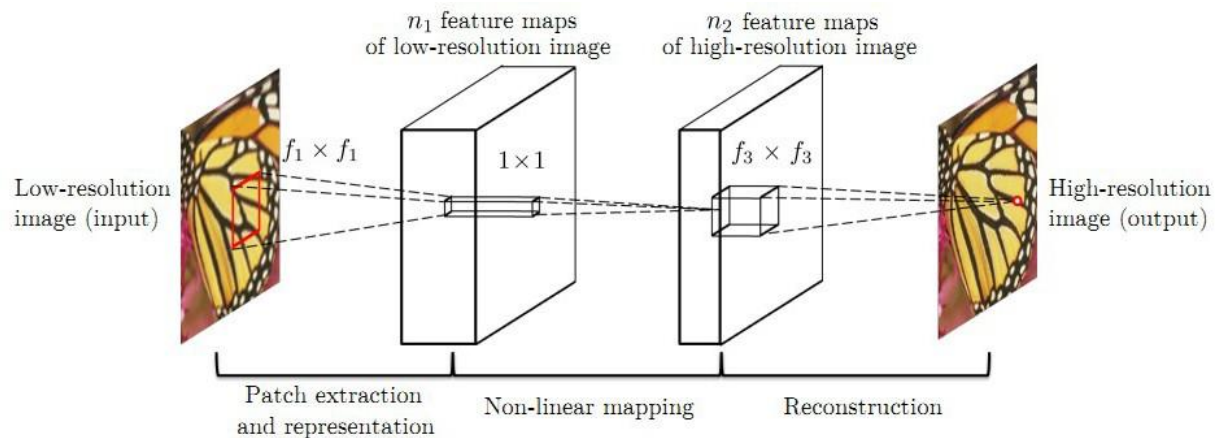


Рис 1.3 – Архітектура мережі SRCNN [6]

Позначимо вихідне зображення як  $Y$ , а зображення з високою роздільністю, з яким буде здійснюватися порівняння –  $X$ . Зображення  $Y$ , незважаючи на відповідність розширенню зображення  $X$ , будемо вважати зображенням низької роздільності. На рис. 1.3 зображення  $Y$  позначено написом «Low resolution image (input)».

В результаті роботи нейронної мережі необхідно отримати зображення  $F(Y)$  об'єктивно більш подібне, ніж спотворене, до вхідного зображення  $X$ . Для цього необхідно знайти функцію  $F$ . Пошук складається з наступних 3-х операцій:

1. Отримання патчів. На рис. 1.3 дана операція позначена написом «Patch extraction and representation». Результатом роботи даної операції є патчі (окремні ділянки зображення, що складаються з декількох пікселів) з вхідного зображення низького розширення  $Y$ , а потім представити кожен патч у вигляді багатовимірного вектора. Ці вектори містять набір карт ознак, розмір яких дорівнює розмірності вектора ( $n_1$  feature maps of low-resolution image на рис. 1.3).

2. Нелінійне відображення. На рис. 1.3 дана операція позначена написом «Non-linear mapping». Результатом операції є нелінійне відображення кожного багатовимірного вектора на інші вектори. Кожен відображений вектор являє патч з високою роздільною здатністю. Ці вектори містять інший набір карт ознак (n2 feature maps of high-resolution image на рисунку рис. 1.3)
3. Реконструкція. На рис. 1.3 позначена написом «Reconstruction». Дана операція використовує патча, отримані з нелінійного відображення використовуються для того, щоб згенерувати зображення високої роздільної здатності  $F(Y)$ . Очікується, що отримане зображення буде об'єктивно схожим з оригінальним зображенням  $X$ .

Отримане в результаті роботи нейронної мережі зображення  $F(Y)$  позначено на рис. 1.3 написом «High-resolution image (output)».

Всього дана модель містить три шари.

Для отримання патчів використовується перший шар нейронної мережі, який можна позначити (1.1):

$$F_1(Y) = \max(0, W_1 * Y + B_1), \quad (1.1)$$

де  $W_1$  – фільтри,  $B_1$  – ваги,  $*$  – операція згортки.

Фільтри  $W_1$  можна позначити через (1.2):

$$W_1 = c \times f_1 \times f_1, \quad (1.2)$$

де  $c$  – кількість каналів зображення  $Y$ ,  $f_1$  – розмір фільтра у просторі.

Фільтри  $W_1$  здійснюють  $n_1$  операцій згортки зображення, і кожна згортка має ядро згортки розміру  $c \times f_1 \times f_1$ . Вихідні дані шару містять карти ознак в кількості  $n_1$ .

Ваги  $B_1$  це  $n_1$ -мірний вектор, кожен елемент якого співставлено з елементом фільтра  $W_1$ .

Як функції активації першого шару використовується ReLU (Rectified Linear Unit, «випрямляч») (1.3):

$$f(x) = \max(0, x), \quad (1.3)$$

де  $x$  – вхідний сигнал.

Другий шар здійснює операцію нелінійного відображення  $n_1$ -мірний векторів на  $n_2$ -мірні вектори. Операція працює лише з фільтрами розміром  $1 \times 1$ .

Операція, яка здійснюється другим шаром ШНМ, математично описується наступним чином в (1.4):

$$F_2(Y) = \max(0, W_2 * F_1(Y) + B_2), \quad (1.4)$$

де параметри  $W_2$  і  $B_2$  аналогічні параметрам для  $F_1$ , але фільтри тепер здійснюють  $n_2$  операцій, а вагами є  $n_2$ -мірний вектор.

Подібно до першого шару в даному випадку в якості функції активації також використовується ReLU.

Третій шар здійснює операцію реконструкції. Його можна представити (1.5):

$$F(Y) = W_3 * F_2(Y) + B_3, \quad (1.5)$$

де  $W_3$  – це фільтри розмірністю  $n_2 \times f_3 \times f_3$ ,  $B_3$  –  $c$ -мірний вектор.

На виході після обробки даних третім шаром виходить зображення з високою роздільною здатністю.

Функція активації даного шару (1.6):

$$f(x) = x, \quad (1.6)$$

де  $x$  – це вхідний сигнал.

В оцінці втрат використовується середня квадратична помилка MSE (Mean Squared Error) (1.7):

$$L(\theta) = \sum_{i=1}^n |F * (Y_i) - X_i|, \quad (1.7)$$

де  $\theta$  – це параметри нейронної мережі  $\{W_1, W_2, W_3, B_1, B_2, B_3\}$ ,  $n$  – кількість зразків в навчальній вибірці.

При аналізуванні даної моделі було проведено тестування для різних значень вхідних даних для нейронної мережі. Було з'ясовано, що найбільш оптимальними є наступні вхідні дані:  $f_1 = 9, f_2 = 1, f_3 = 5, n_1 = 64$  і  $n_2 = 32$  [6].

#### **1.4 Rapid and Accurate Image Super-Resolution**

Переважаючо для інтерполяції зображень застосовуються прості інтерполятори, які знаходять проміжні значення нових пікселів по відомому набору значень пікселів вихідного зображення. В таких випадках застосовуються різні методи для обчислення середніх значень: інтерполяція методом найближчого сусіда, білінійна інтерполяція, кубічний метод, бікубічний метод і т.д. Перелічені методи використовують прості математичні формули. Вони широко використовувалися в різних додатках в силу своєї простоти і швидкодії. Вони абсолютно не адаптуються до змісту зображення, що часто призводить до появи неприємних артефактів – занадто розмитих фрагментів, характерних спотворень [7].

В ході навчання RAISR застосовувалася база зображень, яка одночасно містить однакові зображення високої якості та спотворені. Використовувалися пари маленьких фрагментів зображення для стандартної двократною інтерполяції, тобто фрагменти  $3 \times 3$  і  $6 \times 6$  пікселів. Алгоритм навчання і роботи RAISR відображено на рис. 1.5 [7].

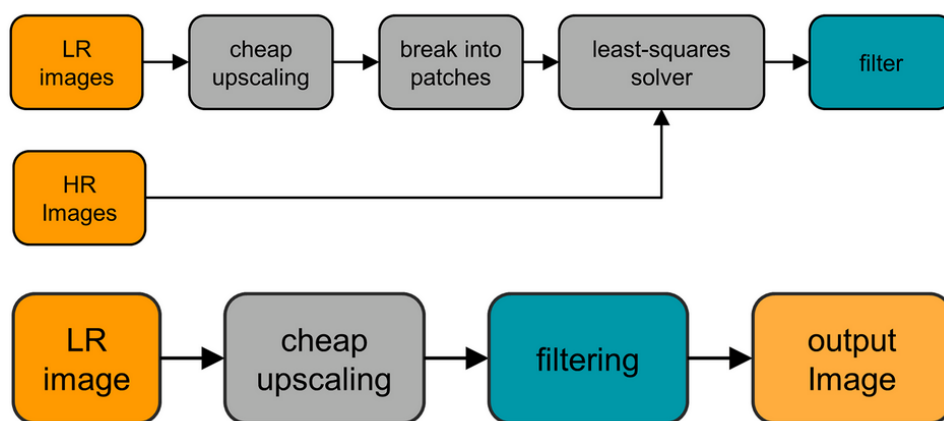


Рис 1.5 – Схема роботи і навчання методу RAISR [7]

Саме навчання відбувається за допомогою наступної послідовності дій:

1. На вхід потрапляє зображення низької роздільності.
2. Його роздільність підвищується за допомогою білінійної інтерполяції.
3. За допомогою фільтрів (з ними можливо ознайомитися в оригінальній роботі [7]) оброблене зображення, яке пройшло білінійну інтерполяцію, ділиться на 4 різних патчі.
4. За допомогою фільтрів оригінальне зображення ділиться на 4 різних патчі.



5. За допомогою отриманих патчів та методу найменших квадратів проводиться регресійний аналіз.

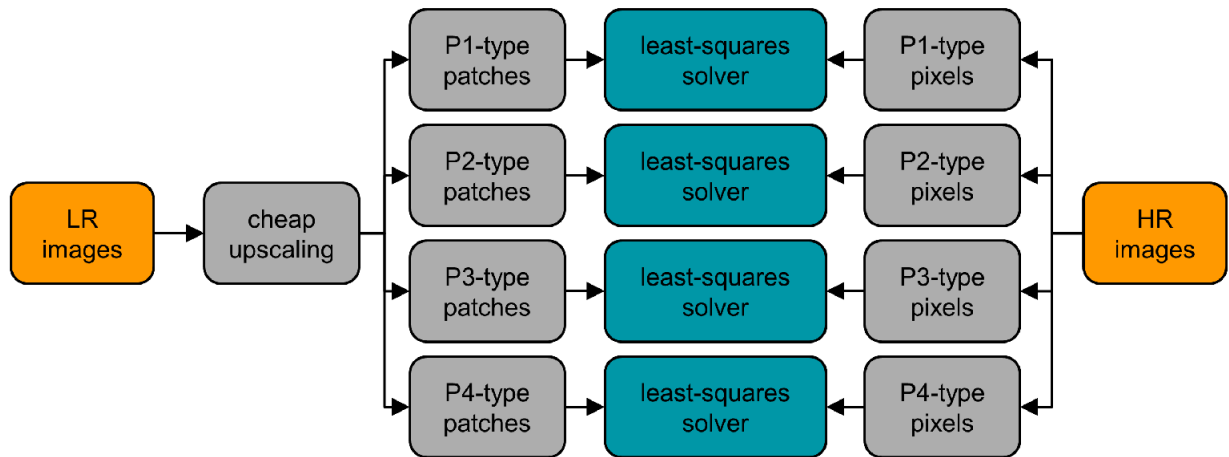


Рис 1.6 – Навчання методу RAISR [7]

У чомусь метод машинного навчання RAISR схожий на навчання нейронної мережі. Але фактично він є адаптацію різних фільтрів стандартної інтерполяції для кожного окремого маленького фрагмента вихідного зображення. Тобто це і є «лінійна інтерполяції», але з покращеннями – без властивих їй артефактів і з адаптацією до змісту зображення.

Порівняльне тестування показало, що такий алгоритм у багатьох випадках працює навіть краще, ніж сучасні методи інтерполяції, засновані на нейронних мережах рис 1.7 [7].

До того ж, такий метод на основі хешування набагато менш ресурсномісткий і більш прийнятний на практиці, ніж навчання і використання нейронної мережі. Різниця в продуктивності настільки велика (10-100 разів), що цю програму можна запускати навіть на звичайних мобільних пристроях, і вона буде працювати в реальному часі [7].

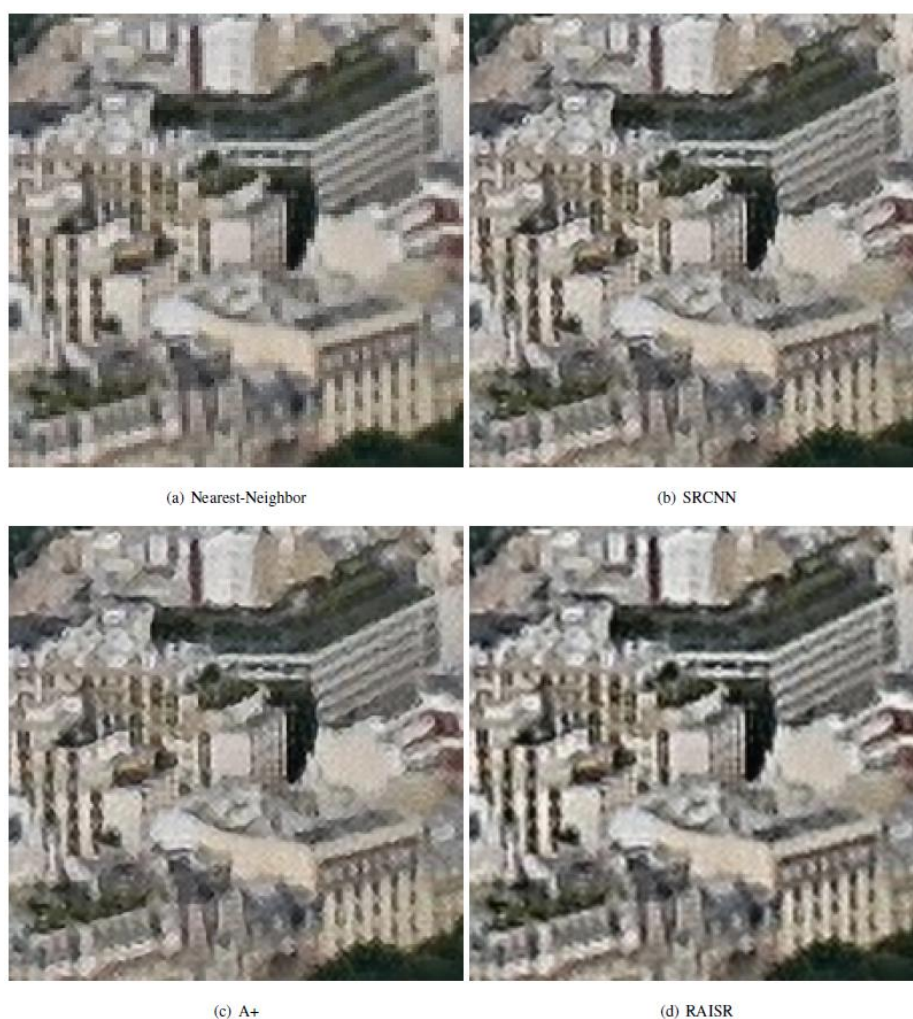


Рис 1.7 – Підсумкове порівняння методів наведених у даному розділі  
а) найближчого сусіда; б) SRCNN; в) A+; г) RAISR; [7]

На сьогодні існує можливість інтегрувати даний метод в сучасні програми інтерполяції зображень на смартфонах, в тому числі в додаток камери на Android, яке виконує інтерполяцію під час цифрового масштабування. Цілком можливо, що Google в першу чергу розробляла даний алгоритм для цієї цілі.

### 1.5 Нейронні мережі

Штучна нейронна мережа є концептуальною моделлю біологічної нейронної мережі і складається з пов'язаних між собою різним чином шарів

штучних нейронів, які організовують загальну активну структуру і функціонально впливають на роботу один одного. У більшості архітектур штучних нейронних мереж, активність нейрона визначається перетворенням зовнішнього сумарного впливу інших нейронів на даний нейрон [1].

Більшості архітектур штучних нейронних мереж притаманні фіксовані функції активації нейронів, а ваги синапсів є параметрами мережі. Деякі входи нейронів є зовнішніми входами сукупної мережі, а деякі виходи нейронів – виходами сукупної мережі.

В основі штучної нейронної мережі лежить гіпотетична структура рис. 1.5, яка приймає сигнал, перетворює його (подібно до біологічних нейронів), і передає іншим нейронам (які роблять те ж саме). Таку гіпотетичну структуру і називають штучним нейроном.

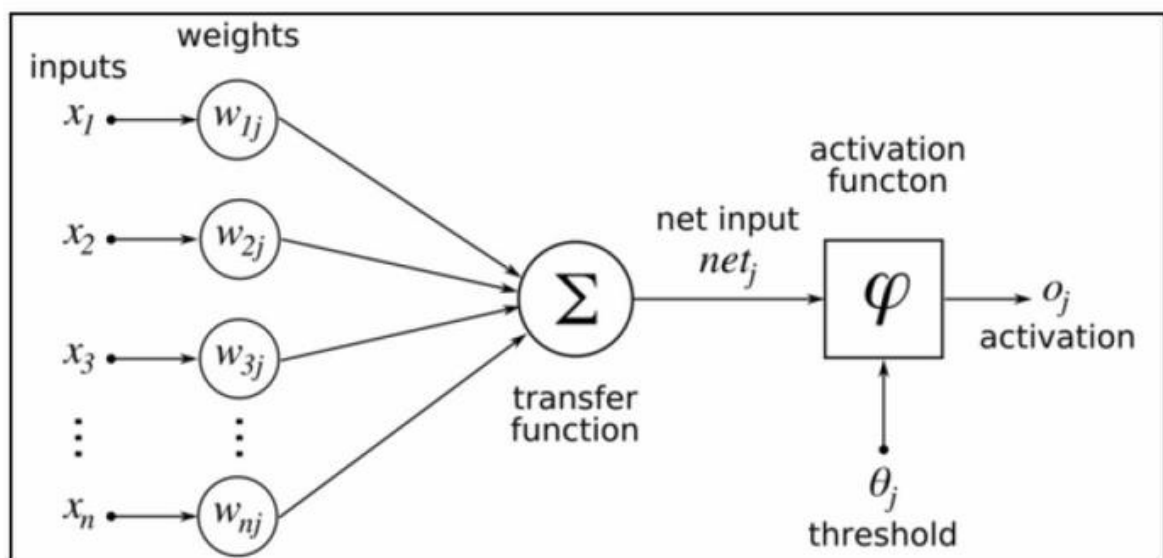


Рис 1.5 – Загальна структура штучного нейрона

Таким чином визначення терміну «нейронна мережа» є наступним: це сукупність взаємодіючих між собою штучних нейронів.

Тож кожен нейрон оброблює сигнал за певним алгоритмом, але також необхідна враховувати, що в біологічних нейронних мережах в обробці інформації бере участь і синапс.

Синапс – місце стику виходу одного нейрона і входу іншого, в якому відбувається посилення або послаблення сигналу.

У кожного біологічного нейрона є тисячі входів. Кожен з них з'єднаний з виходами інших нейронів. В такому випадку на кожен нейрон приходить декілька тисяч синапсів.

Кожен синапс нейронної мережі індивідуальний. Він може або посилювати, або послаблювати сигнал, який проходить через нього. Більш того, з плином часу синапси можуть змінюватися, а це означає, що також буде змінюватися характер зміни сигналу. Якщо правильно підібрати параметри синапсів, то вхідний сигнал, після проходження через нейронну мережу, буде перетворюватися в правильний вихідний сигнал.

Нейронні мережі застосовують для вирішення завдань, алгоритм вирішення яких не відомий.

Крім можливості вирішувати новий клас задач, нейронні мережі мають ряд значних переваг. Усі переваги нейронних мереж є наслідками плюсів біологічних нейронних мереж, оскільки сама модель обробки інформації при проектуванні штучних не змінюється.

Загалом, існує кілька способів графічного зображення нейронних мереж і нейронів. Надалі у роботі штучні нейрони зображені у вигляді кіл рис 1.6.

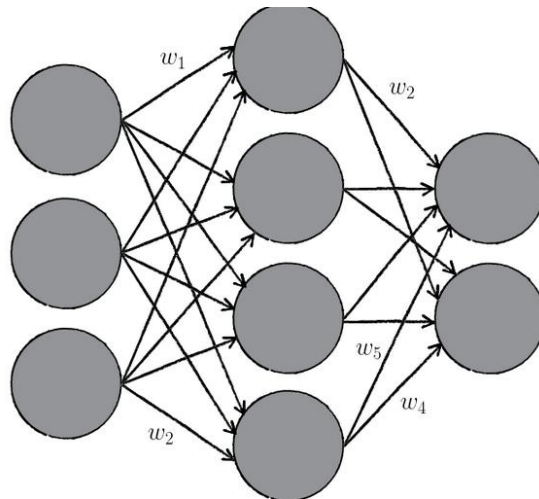


Рис 1.6 – Нейронна мережа з вагами зв'язків

На входи мережі передаються числа для обробки, подібно електронному сигналу який проходить через синапс. Ці значення проходять по мережі і відповідно до архітектури мережі змінюються. На виході мережі отримується результуюче число, яке буде результатом роботи мережі.

Кожен зв'язок нейронної мережі характеризується певним числом, яке називається вагою зв'язку. Сигнал, що пройшов через даний зв'язок, множиться на вагу відповідної зв'язку. Це ключовий момент в концепції штучних нейронних мереж.

На рис. 1.6 кожній чорній стрілці відповідає певна кількість  $w_i$  (вага зв'язку). Коли сигнал проходить по зв'язку, його величина множиться на вагу зв'язку. У кожного  $i$ -го зв'язку є своя власна вага.

У кожного нейрона, в тому числі і у штучного, повинні бути входи, через які він приймає сигнал.

Сигнали, які надходять на входи множаться на свої ваги рис 1.7. Сигнал першого входу  $x_1$  множиться на відповідну вагу  $w_1$ . У підсумку

отримуємо  $x_1 w_1$ . І так до  $n$ -ого входу. У підсумку на останній вхід отримуємо  $x_n w_n$ . Надалі всі добутки передаються у суматор.

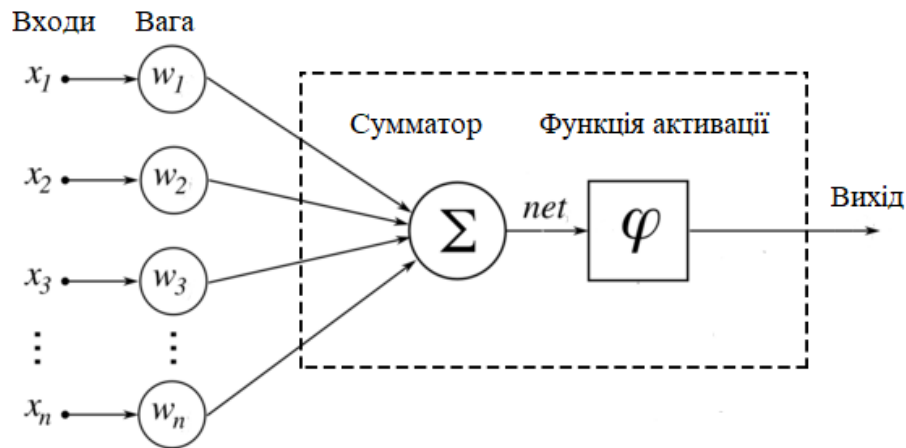


Рис 1.7 – Повна модель штучного нейрона

Суматор підсумовує всі вхідні сигнали, помножені на відповідні ваги, відображено в (1.1):

$$x_1 w_1 + \dots + x_n w_n = \sum_{i=1}^n x_i w_i \quad 1.1$$

Переважно після обробки інформації нейронами, немає сенсу сумувати результати роботи виходу кожного з них, саме тому існує поняття функція активації, яка перетворює зважену суму в якесь число, яке і є виходом нейрона (вихід нейрона позначається змінною out).

Для різних типів штучних нейронів використовують найрізноманітніші функції активації. У загальному випадку їх позначають символом  $\varphi$  [2].

## **1.6 Висновок до першого розділу**

В даному розділі були розглянуті деякі основи, які стосуються обробки зображень, а також види задач в напрямку вирішення яких розвивається програмне забезпечення.

З одного боку, на момент дослідження існує проблема якості зображень, а саме: низька роздільність, шум, розмитість, та не усі перелічені проблеми мають якісні рішення.

З іншого ж боку, останнім часом нейронні мережі набули дуже стрімкого розвитку, в тому числі у напрямку обробки зображень, в даному розділі описана загальна структура штучної нейронної мережі та її елементів. Також була відображена загальна схема нейронної мережі.

Отже, у даному розділі було описано існуючі методи з обробки зображень, деякі з них є стандартними методами інтерполяції, інші ж побудовані з використанням нейронних мереж. Архітектура останніх складається з багатьох рівнів та є досить складною для дослідження.

Грунтуючись на проведеному аналізі, можна зробити висновок, що на сьогодні існує потреба у вдосконаленні існуючих підходів до задачі покращення якості зображень.

## **2. КОМПЛЕКСНА ОЦІНКА ТА ПОКРАЩЕННЯ ЯКОСТІ ЗОБРАЖЕНЬ**

### **2.1 Комплексна оцінка якості зображень**

Надалі у даній науковій роботі буде використовуватися набір зображень (Set5) які прийнято використовувати в задачах обробки, порівняння та аналізу зображень [18].

В межах проведеного дослідження була розроблена методика комплексної оцінки зображень, для створення якої були досліджені статі [11, 12, 13], які стосуються аналізу та покращення зображень та наукову роботу [21], яка порівнює метрики структурної подібності, середню квадратичну похибку та метрику яка вимірює співвідношення між максимумом можливого значення сигналу та потужністю шуму. Жодна з перелічених робіт не дає однозначної відповіді на питання, як краще оцінювати якість обробленого зображення відносно оригінального. Після проведених досліджень в даній дисертаційній роботі було запропоновано нову методику яка включає в себе наступні метрики:

1. PSNR,
2. SSIM,
3. MSE.

Далі у даному розділі наведено опис даних метрик та опис розробленої методики.

Найпростішою метрикою, як з точки зору математики, так і програмування, є MSE. В її основі лежить порівняння двох зображень по кожному пікселю (2.1).



$$MSE(x, y) = \sum_{i=1}^n (x_i - y_i)^2, \quad (2.1)$$

де  $x_i$  –  $i$ -й піксель зображення  $x$ ,  $y_i$  –  $i$ -й піксель зображення  $y$ ,  $n$  – загальна кількість пікселів.

Якщо при порівнянні двох зображень значення MSE дорівнює нулю, це говорить про те, що зображення повністю ідентичні. Чим вище значення метрики MSE, тим більше відмінностей між двома зображеннями.

Пікове співвідношення сигналу до шуму (PSNR) показує співвідношення між максимумом можливого значення сигналу і потужністю шуму, що спотворює значення сигналу. Так як багато сигналів мають широкий динамічний діапазон, PSNR прийнято зазвичай вимірювати за логарифмічною шкалою в децибелах. Чим більше значення PSNR, тим більша схожість мають порівнювані зображення.

Можна визначити через середню квадратичну помилку (MSE). Формула має такий вигляд:

$$PSNR = 10 \log_{10} \left( \frac{MAX_I^2}{MSE} \right) = 20 \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right), \quad (2.2)$$

де  $MAX_I$  – максимальне значення пікселя зображення.

Наукові роботи які проводили кореляцію між метрикою PSNR та експертною оцінкою стверджують, що показник PSNR достатньої (приблизно 6/10 балів) для людини якості повинен бути більшим 20db, а задовільною (приблизно 8.5/10 балів) якістю для зображення формату .jpeg вважається 35db [22, 23].

Індекс структурної подібності (SSIM) – порівнює два вхідні зображення. Але на відміну від інших метрик даний метод враховує

«сприйняття помилки», завдяки тому що враховується структурна зміна інформації. Пікселі мають сильний взаємозв'язок, коли вони знаходяться поруч у просторі [20]. Дана метрика вимірюється у відсотках. Математично метрика SSIM для зображення  $x$  та  $y$  розміру  $N \times N$  описана у (2.3) [20].

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (2.3)$$

де змінні:  $\mu_x$  – середнє  $x$ ;  $\mu_y$  – середнє  $y$ ;  $\sigma_x^2$  – дисперсія  $x$ ;  $\sigma_y^2$  – дисперсія  $y$ ;  $\sigma_{xy}$  – коваріація  $x, y$ ;  $c_1 = (k_1L)^2$ ,  $c_2 = (k_2L)^2$ ;  $L$  – динамічний діапазон пікселів;  $k_1$  – константа зі значенням 0,01;  $k_2$  – константа зі значенням 0,03.

Для комплексної оцінки зображення в даній дисертаційній роботі пропонується ввести змінну  $p_{\text{ком}}$ , яка буде включати усі наведені вище метрики, що нормалізовані та обчислювати комплексну оцінку за (2.4).

$$p_{\text{ком}} = \frac{p_1k_1 + p_2k_2 + p_3k_3}{3}, \quad (2.4)$$

де змінні:  $p_1$  – нормалізована оцінка PSNR у діапазоні від 0 до 1;  $p_2$  – нормалізована оцінка MSE у діапазоні від 0 до 1;  $p_3$  – нормалізована оцінка SSIM у діапазоні від 0 до 1 (Потреба в нормалізації, адже даний показник вимірюється у відсотках);  $k_1, k_2, k_3$  – вагові коефіцієнти від 0 до 3, але  $k_1 + k_2 + k_3 \leq 3$ .

Таким чином, комплексна оцінка враховує 3 метрики. Вагові коефіцієнти можливо змінити, якщо необхідно, щоб певна метрика більше впливала на кінцеву оцінку у специфічній задачі або на специфічному наборі даних.

## 2.2 Архітектура нейронної мережі

Після дослідження робіт [10, 11, 12, 13] та методів, що згадані у попередньому розділі, стає зрозуміло сконцентровані на підвищенні якості зображень за допомогою згорткових нейронних мереж, стає зрозуміло, що різні способи на основі нейронних мереж в певній мірі вирішують проблему низької роздільності зображення, але інколи для використання даних способів необхідно мати великі обчислювальні потужності, які виходять за рамки користувацьких пристроїв.

Також вони в певній мірі вирішують проблему втрати частини сигналу. Але переважна більшість з перерахованих способів працюють лише за умови наявності набору вхідних даних для навчання нейронної мережі, який складається з великої кількості зображень високої якості та тих самих зображень, які пройшли процес зниження розширення.

У роботі [14] доведено що нейронні мережі, які пройшли тренування на хибних наборах даних, можуть в деяких випадках показати кращі результати, ніж методи, які пройшли навчання на звичних наборах даних [12]. Тому в межах даної роботи було вирішено розробити штучну згорткову нейронну мережу, яка не потребує навчання на вхідних наборах даних, а замість того використовує згортку та фільтри, але яка пізніше може використовуватися з навчанням, а також сконцентрувати увагу на кількості рівнів штучної нейронної мережі та впливі аргументів які передаються до нейронної мережі на кінцевий результат.

Для остаточних експериментів за основу архітектури нейронної мережі було обрано архітектуру Hourglass U-Net, з ціллю отримати найкращий результат, так як дана архітектура нейронної мережі уже застосовувалася для обробки цифрових зображень з шумом [15].

Загальна архітектура мережі зображена на рис. 2.1 [19].

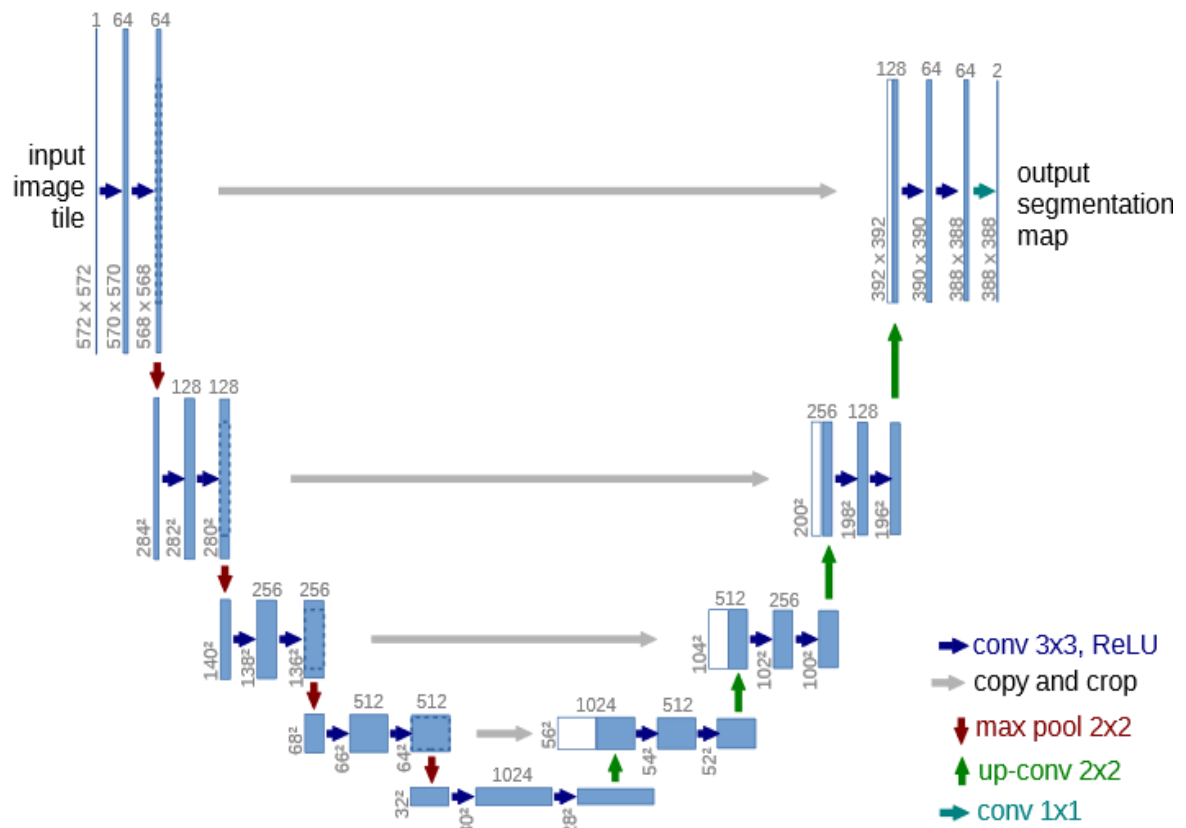


Рис 2.1 – Схематичне зображення архітектури нейронної мережі Hourglass U-Net яка була розроблена для обробки зображень з шумом [19]

Архітектура нейронної мережі, що пропонується у даній дисертаційній роботі, представлена у вигляді послідовності дій рис. 2.2. Підґрунтя для використання чотирьох шарів нейронної мережі описано у розділі 4.

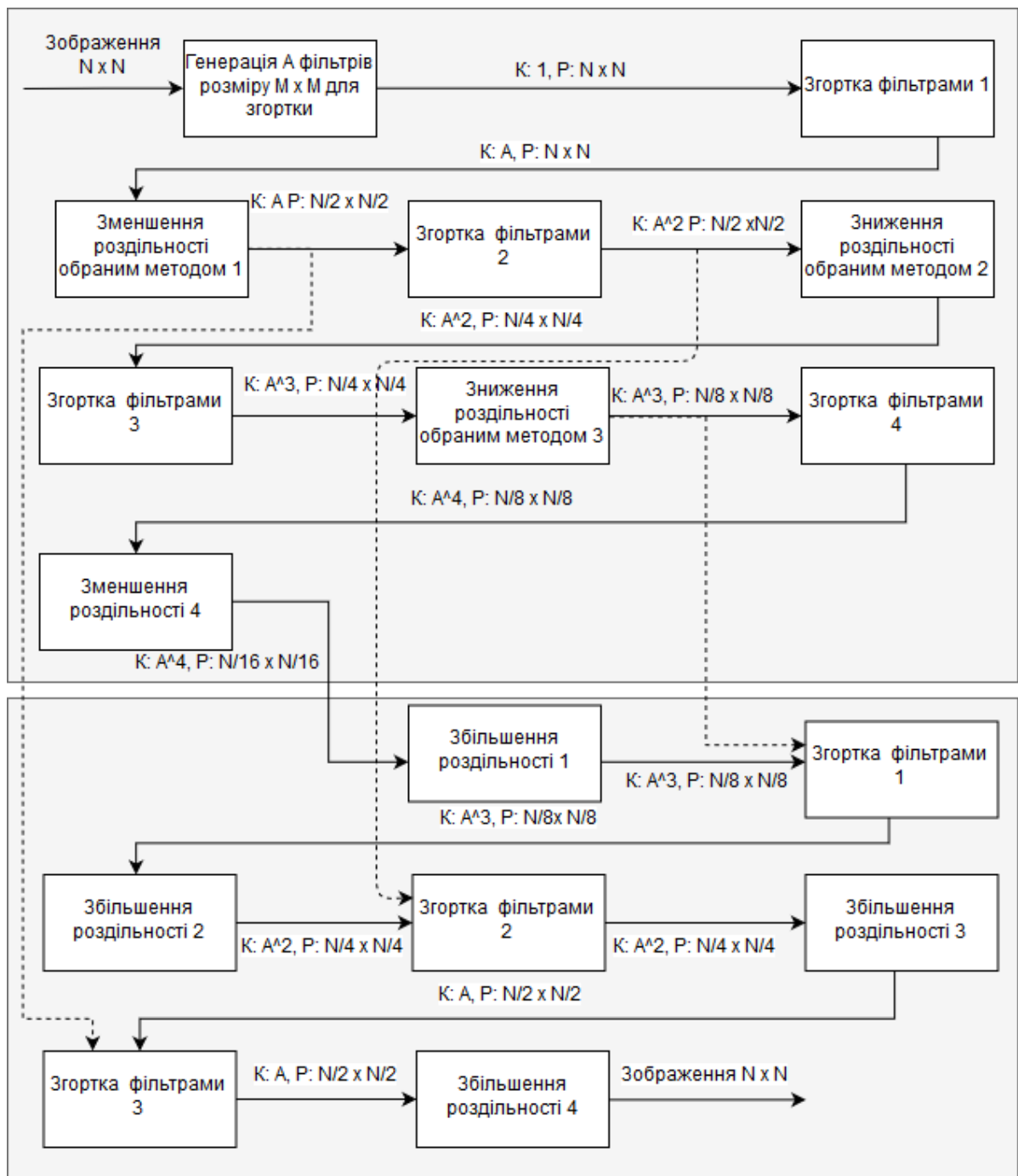


Рисунок 2.2 – Загальна архітектура розробленої нейронної мережі,  
 К – кількість карт ознак, Р – розмір.

Дана нейронна мережа приймає на вхід зображення з роздільністю  $N \times N$ , після чого відбувається генерація фільтрів для згортки розміру  $M \times M$ . Для того, щоб при згортці фільтр не виходив за межі зображення, або ж

згенерованих карт ознак, вони доповнюються за допомогою нулів або алгоритмів заповнення відступу. Для згортки необхідно обрати кількість та розмірність фільтрів, а також їх наповнення. Для зменшення роздільності необхідно обрати відповідний метод або алгоритм. В межах даної роботи розглядалися Average Pool, Max Pool, фільтр Ланцоша (формула 3.1) та метод найближчого сусіда. Для збільшення роздільності розглядалася лише білінійна інтерполяція. В якості функції активації використовується ReLU. Результатом роботи нейронної мережі є зображення розміром  $N \times N$ .

$$L(x) = \begin{cases} \text{sinc}(x)\text{sinc}\left(\frac{x}{a}\right), & -a < x < a \\ 1, & x = 0 \\ 0 & \end{cases}, \quad (3.1)$$

Функція активації ReLU (rectified linear unit). Має наступн визначення (3.2):

$$f(x) = \max(0, x), \quad (3.2)$$

Обчислення сигмоїд і гіперболічного тангенса вимагає виконання ресурсоемних операцій, таких як зведення в ступінь.

Застосування ReLU істотно підвищує швидкість збіжності стохастичного градієнтного спуску (в деяких випадках до 6 разів) в порівнянні з сигмоїдами і гіперболічним тангенсом. Вважається, що це обумовлено лінійним характером.

Застосування функції активації ReLU дозволяє отримати наступні переваги:

1. Біологічна правдоподібність: функція є односторонньою на відміну від центрально симетричного гіперболічного тангенса.
2. Розріджена активація: у мережі в якій була проведена ініціалізація за допомогою функції `random()`, тільки близько 50 % прихованих

елементів активуються (мають не нульове, або більше за нього значення).

3. Краще градієнтне поширення: рідше виникає проблема зникання градієнту у порівнянні з сигмоїдальною передавальною функцією, яка може виникнути в обох напрямках.
4. Швидкість обчислення: необхідні тільки функції порівняння, додавання та множення.

Проте функція активації ReLU має такі небажані властивості:

1. ReLU є нерегулярною у нулі, проте, її похідна існує в усіх інших точках. Також можна на власний розсуд задати значення у нулі – 0 або обрана константа.
2. Наявність «мертвих» зон: може трапитись так, що нейрони будуть переведені у стан, в якому вони стануть неактивними для всіх вхідних значень. Для такого стану відсутнє зворотне поширення градієнту, яке проходить через нейрон і тому в такому стані нейрон потрапляє у стан який надалі неможливо змінити. Це один з різновидів проблеми зникання градієнту.

ReLU з «витоком» (Leaky ReLU) є одним з кращих способів, які використовують, щоб вирішити описану вище проблему звичайних ReLU. Звичайний ReLU на інтервалі  $x < 0$  дає на виході нуль, в той час як LReLU має на цьому інтервалі невелике від'ємне значення (кутовий коефіцієнт близько 0,01). Тобто функція для LReLU має вигляд  $f(x) = \alpha x$  при  $x < 0$  і  $f(x) = x$  при  $x \geq 0$ , де  $\alpha$  – мала константа. Деякі дослідники повідомляють [19] про успішне застосування даної функції активації, але результати не завжди стабільні.

### 2.3 Спосіб покращення якості зображень

Розроблений спосіб ґрунтується на використанні розробленої нейронної мережі та комплексної оцінки якості зображень з цілметою отримання зображення об'єктивно кращої якості, ніж вхідне з використанням фільтрів. Оскільки з кожною ітерацією обробки зображення у нейронній мережі на даний момент однаково можливе як підвищення комплексної оцінки, так і її зниження, це твердження також є справедливим для метрики PSNR, яка використовується у більшості наукових робіт.

Було вирішено проводити 15 ітерацій обробки зображень розробленою нейронною мережею (оскільки за результатами проведених досліджень лише в 4% випадків нейронна мережа показує кращий результат на ітераціях з 15 по 100), та між ними обчислювати  $p_{\text{ком}}$ , з вагами  $k_1 = 2, k_2 = 1, k_3 = 0$ , оскільки, як згадувалося вище, PSNR частіше зустрічається у наукових роботах, але SSIM показує оцінку, яка частіше виявляється ближчою до експертної [21], MSE переважно застосовується у математичних задачах, а не у випадках коли необхідно передбачити реакцію людини. В випадку, якщо обидві метрики дадуть вищу оцінку одному методу, буде обрано зображення, якому відповідають дані оцінки, в іншому випадку користувач на виході зі способу отримає два зображення, з найвищою оцінкою PSNR та  $p_{\text{ком}}$ .

Засновуючись на результатах проведених експериментів, частина з яких відображена у розділі 4.2, як найбільш ефективні було обрано аргументи та фільтри для способу:

1. Фільтр згортки розміру 3 x 3, кожен елемент якого лежить в межах від 0 до 1, але фільтр необхідно нормалізувати, щоб сума усіх елементів фільтра не перевищувала 1.



2. Функцію Max Pool розміру 3 x 3 з кроком 2, для зменшення роздільності.
3. Білінійна інтерполяція розміру 2 x 2 для підвищення роздільності.
4. Функція активації LReLU.

Отже запропонований спосіб полягає в ітеративному використанні розробленої архітектури нейронної мережі з фільтром згортки розміру 3 x 3, функцією Max Pool для зменшення роздільності, Білінійної інтерполяції для підвищення роздільності та функції активації LReLU.

#### **2.4 Висновок до другого розділу**

У даному розділі було проведено аналіз існуючих метрик для оцінки якості зображень відносно оригіналу а також описано власну методику комплексної оцінки якості зображень. Також було розглянуто математичне підґрунтя метрик MSE, SSIM, PSNR.

Було проаналізовано наукові роботи, які містять у собі способи обробки зображень за допомогою нейронних мереж, на основі даних робіт та експериментів, що описані у розділі 4, було вирішено за основу нейронної мережі взяти архітектуру U-Net Hourglass, оскільки в роботах які були проаналізовані вказано, що дана нейронна мережа була створена для подібних задач.

Також було описано розроблену архітектуру нейронної мережі і перераховано методи та алгоритми, які в ній використовуються, або можуть використовуватись.

На відміну від існуючих способів інтерполяції, що не використовують нейронні мережі, розроблений спосіб показує вищі результати об'єктивних метрик.

### 3. ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СПОСОБУ

#### 3.1 Необхідні засоби програмної реалізації

Для вирішення різних завдань машинного навчання і обробки зображень використовуються різні парадигми програмування. Також існує велика кількість різних бібліотек, які частково надають доступ до функцій, які є необхідними для реалізації способу.

Запуск, тестування та виконання усіх функцій програмного забезпечення виконувались на комп'ютері Google cloud з конфігурацією відображеною в таблиці 3.1.

Таблиця 3.1 – конфігурація комп'ютера Google cloud на якому відбувався запуск програмного забезпечення

Назва віртуального комп'ютера в системі google cloud	Кількість ядер CPU	Кількість RAM	GPU
n1-highcpu-8m	8	7.20GB	Tesla V100

Для програмної реалізації запропонованого способу було використано мову програмування Python 3. На цей вибір вплинули перераховані нижче фактори:

1. Дана мова є однією з найбільш популярних мов програмування в академічному середовищі при використанні завдань, пов'язаних з машинним навчанням і інтелектуальною обробкою даних [8]. Таким чином, програмна реалізація з використанням цієї мови буде зрозуміла і доступна широкому колу дослідників.
2. Для даної мови програмування існує велика кількість бібліотек, які спрощують роботу зі складними алгоритмами і методами машинного

навчання. Також для Python існує безліч бібліотек для роботи з зображеннями. Варто також відзначити легкість встановлення бібліотек за допомогою вбудованих в мову менеджерів пакетів(рір).

3. Python є мовою програмування високого рівня, яка дозволяє швидко проектувати прототипи програм, що безпосередньо дозволяє скоротити час розробки.
4. Python підтримує модульність за рахунок того, що кожен файл в проєкті може бути виконаний як окремий скрипт, а також бути використаним при імпорті окремих методів.
5. Однак у Python є і мінуси. Через те, що він належить до мов програмування, які оброблюються інтерпретатором, програмні засоби, реалізовані на ньому, як правило, працюють повільніше аналогічних алгоритмів реалізованих на компільованих мовах програмування. Але цей мінус частково компенсується використанням бібліотек, в яких була проведена оптимізація з використанням коду, на інших мовах.

При написанні програмної реалізації використовувалася версія мови 3.6.6. була найактуальнішою стабільна версією мови на момент початку написання програми.

Для програмної реалізації було вирішено використовувати такі основні бібліотеки:

1. NumPy.
2. OpenCV.
3. PyTorch.

Також використовувалися різні допоміжні пакети, такі, як shutil для роботи з файлами і папками.

NumPy – це бібліотека для мови Python, що додає підтримку великих багатовимірних масивів і матриць, разом з великою бібліотекою високорівневих математичних функцій для операцій з цими масивами. Крім роботи з масивами, в даній бібліотеці реалізовано функції лінійної алгебри, перетворення Фур’є, генерації випадкових чисел тощо.

Як уже згадувалося раніше в даному розділі, Python, як правило, обробляє складні математичні алгоритми повільніше, ніж мови програмування, які компілюються. Автори NumPy частково вирішили цю проблему за допомогою реалізації деяких функцій на двох мовах з компілятором: C і Fortran.

При створенні програмної реалізації використовувалася версія NumPy 1.13.3.

PyTorch – бібліотека машинного навчання для мови програмування Python. Бібліотека була створена на базі Torch. Використовується для розпізнавання людської мови або образів на зображеннях, що дозволяє застосовувати її для покращення зображень.

OpenCV – це бібліотека з відкритим кодом, що містить набір типів даних, функцій і класів для обробки зображень алгоритмами комп’ютерного зору. Написана і оптимізована на C / C ++, також має інтерфейси для Python, Java, Ruby, Lua та інших мов [8].

Ця бібліотека являє собою набір модулів. Кожен з цих модулів пов’язаний з певною областю комп’ютерного зору.

Алгоритми, реалізовані в бібліотеці, протестовані і оптимізовані, а значить, подібно бібліотеці NumPy, дозволяють частково зменшити втрати продуктивності, пов’язані з використанням інтерпретатора Python.

У програмній реалізації була використана версія бібліотеки 3.3.0.

Оскільки OpenCV, в основному, реалізує тільки базові операції з зображеннями, її можна розглядати як низькорівневу бібліотеку. Для вирішення завдань машинного навчання при обробці зображень її, як правило, недостатньо. Необхідно використовувати допоміжні бібліотеки.

Структуру програми складають незалежні модулі (папки), які в свою чергу містять один або кілька скриптів. Всі модулі звертаються до одного джерела даних, розташування якого, разом з іншими настройками і глобальними змінними, зберігається в окремому файлі конфігурації.

Програмне забезпечення було написане для операційної системи Linux, проте завдяки платформ використаних інструментів, вона теоретично може бути запущена на операційних системах Microsoft.

### **3.2 Модуль комплексної оцінки зображень**

Даний модуль складається з одного скрипта compare.py, в якому реалізовані метрики порівняння зображень PSNR, MSE і SSIM та розроблену комплексну оцінку. Зміст скрипта представлено в додатку 1.

Незважаючи на те, що існують бібліотеки, в яких вже є готові реалізації заміру перерахованих вище метрик, спроби знайти бібліотеку, в якій були присутні готові рішення одночасно для всіх трьох метрик не були успішними. З цієї причини, а також щоб уникнути створення нових залежностей від сторонніх бібліотек, було прийнято рішення написати власну реалізацію цих метрик.

У цьому модулі були реалізовані функції getMSE, getPSNR, getSSIM, getPOWN в які передаються масиви, отримані за допомогою функції

бібліотеки OpenCV `cv2.imread`. Перераховані вище функції повертають значення відповідної метрики.

Модуль порівнює оригінальні зображення, що зберігаються в `data/test` з зображеннями, отриманими в результаті роботи способу, тобто розробленого програмного засобу з використанням штучної нейронної мережі. Шлях до папок зображеннями вказуються в файлі конфігурації.

Отримані результати зберігаються у вигляді csv-файлів. Даний формат дозволяє здійснювати швидке завантаження результатів в різні табличні редактори для їх подальшого аналізу та виведення. Також крім цього відбувається друк в вікно консолі результатів усіх метрик.

У цьому модулі використані бібліотеки OpenCV і NumPy для проведення математичних операцій над зображеннями. OpenCV також використовується для читання зображень. Крім цього, був використаний стандартний пакет Python `csv` для збереження результатів порівняння зображень у вигляді csv-файлу.

Надалі у порівняннях з іншими методами, способами та алгоритмами буде відображено лише показник PSNR, адже, переважно, дослідники не включають інші об'єктивні показники якості зображень в наукові роботи, по причині гіршої кореляції з експертною оцінкою.

### **3.3 Розроблене програмне забезпечення**

На основі проведених досліджень розроблено програмного забезпечення засобами мови програмування Python, яка отримує на вхід зображення, розширення якого було знижено в 4 рази з оригінального, а потім видає зображення, яке є більш подібним до оригінального. Діаграма класів нейронної мережі зображена на рисунку рис. 3.2.

Загальна послідовність роботи програмного забезпечення (частина з нейронною мережею) написана мовою Python в додатку 2, щоб більш детально зрозуміти архітектуру програмного забезпечення, необхідно ознайомитися з архітектурою Hourglass U-Net, яка була згадана у другому розділі [19].

В класі UNet приймаються аргументи нейронної мережі, які вказують на фільтр згортки діапазон значень його елементів, спосіб зменшення роздільності та спосіб її підвищення а також ініціалізуються по чотири функції down, up, з різними фільтрами, відповідно в функціях down відбувається зниження розширення вхідного для збору карт ознак зображення методом проходження по зображенню фільтрами, а в функціях up розгортка зібраних даних в нове, більш якісне зображення.

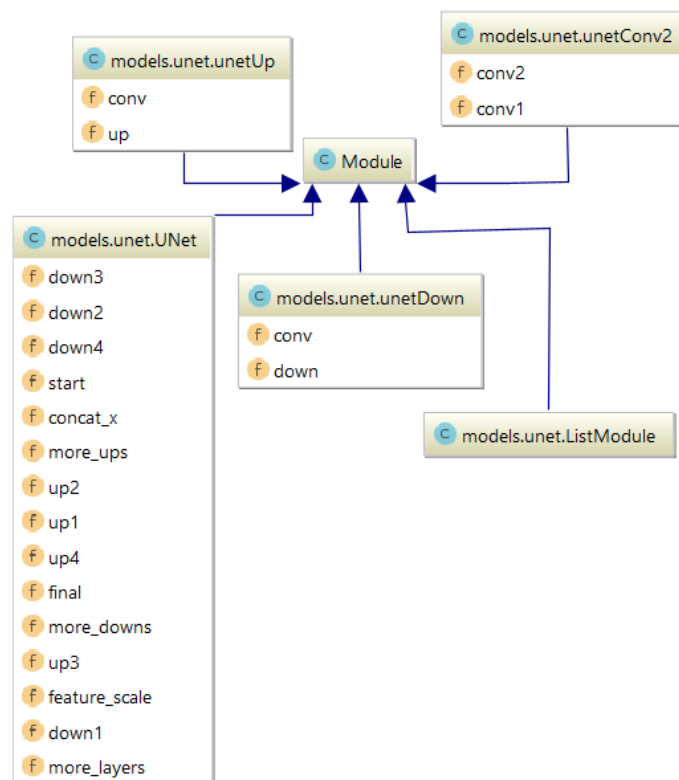


Рис 3.2 – Діаграма класів розробленого програмного засобу

Клас `unetUp` відповідає за підвищення роздільності.

Клас `unetDown` відповідає за зменшення роздільності.

На вхід до програмного забезпечення передаються зображення, роздільність якого методами фільтрації було зменшено в 4 рази, на виході отримуємо зображення приближене розробленим способом до оригіналу.

### **3.4 Висновок до третього розділу**

В даному розділі були розглянуті засоби, які було використано для реалізації програмного засобу та тестування способу. Також наведена діаграма класів розробленого програмного забезпечення і наведено опис додатково розроблених модулів.

В розробленому програмному засобі використовується архітектура нейронної мережі U-Net, функція активації `LeakyReLU`, метод `max pool` та білінійна інтерполяція.

Для роботи даної програмної реалізації необхідна установка мови програмування Python версії 3.6 або вище.

Також необхідно встановити наступні бібліотеки та їх залежності:

1. `OpenCV`.
2. `NumPy`.
3. `PyTorch`.

В результаті було розроблено модуль комплексної оцінки якості зображень та нейронну мережу, яка підвищує роздільність зображень.



## **4. РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ**

### **4.1 Методика оцінювання**

Для коректного порівняння якості роботи алгоритмів і нейронних мереж необхідно використовувати об'єктивні метрики. Вхідними даними для кожного методу буде служити зменшене зображення. Отримане в результаті роботи методів зображення порівнюється з оригіналом на предмет розбіжностей. Тобто чим більше результат схожий на оригінал, тим вище якість роботи алгоритму. Порівняння буде відбуватися за метрикою PSNR та  $p_{\text{ком}}$ , яку було запропоновано у розділі 2 (п. 2.1) даної дисертаційної роботи.

### **4.2 Дослідження для визначення кількості шарів нейронної мережі та аргументів мережі**

Для того, щоб визначити необхідну кількість шарів нейронної мережі обраної архітектури, було по чергово розроблено та протестовано нейронну мережу архітектури U-Net Hourglass з одним, двома, трьома, чотирма, п'ятьма та шістьма шарами. Як метод зменшення роздільності використовувався метод найближчого сусіда, для збільшення білінійна інтерполяція, фільтр згортки розміром 3 x 3, елементи якого були ініціалізовані випадковим чином. Після даного дослідження була обрана архітектура з чотирма шарами ШНМ.

Кращі результати метрики PSNR та  $p_{\text{ком}}$  зображення зебри з Set5 [18] наведено у таблиці 4.1.

Також було проведено тестування фільтрів в межах розробленої нейронної мережі і найкращі результати показала комбінація згортки 4ма фільтрами згортки розміру 3 x 3 за ініціалізацією випадковими числами та подальшою нормалізацією, зменшення роздільності зображення методом

max pool, збільшенням роздільності методом білінійної інтерполяції, порівняння у таблиці 4.2.

Таблиця 4.1 – Результати метрик при обробці зображення з Set5 в залежності від кількості шарів ШНМ

Кількість шарів ШНМ	PSNR	$p_{\text{ком}}$
1	13.86db	0.24
2	15.88db	0.261
3	17.54db	0.27
4	22.01db	0.46
5	20.06db	0.31
6	19.76b	0.29
Результати для зображення, роздільність якого в 4 рази менше оригіналу	15.86db	0.26

Таблиця 4.2 – Результати метрик для різних фільтрів згортки, методів підвищення роздільності та зменшення роздільності

Метод зменшення розширення; збільшення; розмір фільтра згортки	PSNR	$p_{\text{ком}}$
Найб. сус.; білінійна інтерполяція; 3 x 3	22.01db	0.46
Махр pool 2 x 2; білінійна інтерполяція; 3 x 3	25.91db	0.514

Таблиця 4.2 – Продовження

Мах pool 3 x 3; білінійна інтерполяція; 3 x 3	26.21db	0.53
Ядро Ланцоша; білінійна інтерполяція; 3 x 3	25.71db	0.51

### 4.3 Результати роботи програмного засобу

Нижче у даному розділі наведено два зображення зебри: рис. 4.1, рис. 4.2). Перше зображення низької якості, що було отримано шляхом зменшення розширення оригінального в 4 рази. Друге зображення є результатом обробки першого розробленим способом після двохсот ітерацій.



Рис 4.1 – Зображення яке було отримано з оригінального зображення високої якості шляхом зменшення розширення в 4 рази, PSNR 15.86,

$$p_{\text{ком}} = 0.26$$

Як візуально, так і за показниками об'єктивних метрик помітно, що зображення на рис 4.2 є кращим ніж зображення на рис 4.1.



Рис 4.2 – Зображення яке було отримано в результаті обробки зображення розробленим способом, PSNR 25.71,  $p_{\text{ком}}=0.51$

#### **4.4 Візуальне порівняння з іншими способами**

Наділі у даному розділі відображені результати (Рис. 4.3) покращення якості зображень різними засобами, такими як bicubic upsampling, який є представником звичайної інтерполяції, розробленого способу, та методу SRResNet, який вимагає використання професійних графічних прискорювачів та великих наборів даних для навчання.

У кінці розділу 4.4 наведено таблиці з об'єктивною характеристикою PSNR, яка дозволяє оцінити роботу засобів.

#### **4.5 Порівняння результатів роботи за показником PSNR**

В даному підрозділ приведено порівняння розробленого способу з існуючими. Таблиця порівняння розробленого способу з bicubic interpolation та SRResNet за показником PSNR (Табл. 4.3, Табл 4.4), а також, відображені на діаграмі (Рис. 4.4).



Рис 4.3 – Зліва на право розташовані зображення, перший стовбець містить оригінальні зображення, другий – оброблені зображення за допомогою bicubic upsampling, третій з зображеннями обробленими розробленим способом, четвертий – оброблені зображення за допомогою SRResNet

Також у таблицях 4.5, 4.6 наведено результати порівняння розробленого способу та bicubic interpolation за комплексною оцінкою.

Таблиця 4.3 – Порівняння результатів роботи способів bicubic interpolation, SRResNet, розробленого способу за метрикою PSNR

	Дитина	Птах	Метелик	Голова	Жінка
Розд. менше оригіналу в 5 рази	28.16db	25.67db	17.82db	25.94db	23.18db
Bicubic	31.78db	30.2db	22.13db	31.34db	26.75db
SRResNet	33.66db	35.1db	28.41db	32.73db	30.6db
Розроблений спосіб	31.99db	31.8db	26.23db	31.43db	29.1db

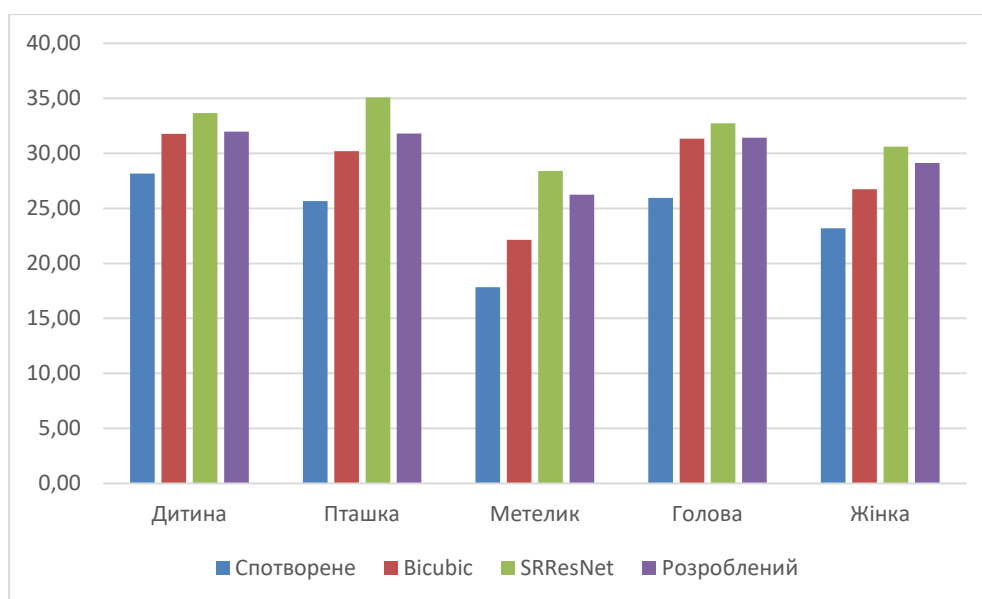


Рис 4.4 – Діаграма з показниками метрики PSNR з таблиці 4.3

Таблиця 4.4 – Відсоткове покращення якості у метриці PSNR відносно вхідного зображення після обробки розробленим способом та способами bicubic interpolation і SRResNet

	Дитина	Птах	Метелик	Голова	Жінка
Bicubic	12,86%	17,65%	24,19%	20,82%	15,40%
SRResNet	19,53%	36,74%	59,43%	26,18%	32,01%
Розроблений спосіб	13,60%	23,88%	47,19%	21,16%	25,54%

Таблиця 4.5 – Порівняння результатів роботи способів bicubic interpolation і розробленого способу за комплексною оцінкою

	Дитина	Птах	Метелик	Голова	Жінка
Розд. в 4 рази менше оригіналу	0.47	0.42	0.25	0.44	0.43
Bicubic	0.59	0.55	0.43	0.59	0.51
Розроблений	0.61	0.608	0.53	0.6	0.56

Таблиця 4.6 – Відсоткове покращення якості за комплексною оцінкою відносно вхідного зображення після обробки способами bicubic interpolation та розробленого способу

	Дитина	Птах	Метелик	Голова	Жінка
Bicubic	25,53%	30,95%	72,00%	34,09%	18,60%
Розроблений	29,79%	44,76%	112,00%	36,36%	30,23%

У кожному з експериментів розроблений спосіб перевищив результат методу Bicubic Interpolation за комплексною оцінкою та метрикою PSNR, але не перевершив результати способу SRResNet.

#### **4.6 Висновок до четвертого розділу**

У даному розділі відображено результати досліджень, на основі яких приймалися рішення щодо особливостей реалізації програмного забезпечення, а також результати роботи розробленого програмного забезпечення, яке включає в себе комплексну оцінку та покращення якості зображень за допомогою нейронних мереж, а крім того приведено порівняння розробленого програмного забезпечення з іншими засобами.

Розроблений спосіб має перевагу над швидкодійним методом інтерполяції Bicubic Upsampling за метрикою PSNR, та перевагу над методом SRResNet, який також побудований з використанням нейронної мережі в відсутності необхідності навчати нейронну мережу на великих вхідних наборах зображень високої та низької якості.

## **5. РОЗРОБЛЕННЯ СТАРТАП ПРОЕКТУ**

### **5.1 Опис проблеми та дерево проблем.**

#### **5.1.1. Анотація проекту**

Проект спрямований на розробку та тестування способу комплексної оцінки та покращення якості зображень з використанням нейронних мереж. Розробка передбачає винайдення способу комплексної оцінки та покращення якості зображень з використанням нейронних мереж. Тестування передбачає аналіз та обробку зображень програмними засобами для подальшого використання розробленого методу в межах даного рішення.

#### **5.1.2. Опис проблеми, на розв'язання якої спрямований проект**

Щодня кількість цифрової техніки зростає, саме тому кількість систем, методів, алгоритмів які спрямовані на аналіз та обробку цифрових сигналів. Підсилює цей процес інтеграція сучасних комп'ютерних та телекомунікаційних технологій. Особливого розвитку в умовах сьогодення набувають методи цифрової обробки зображень, оскільки вони становлять значну частину загального трафіку мультисервісних мереж. Вирішення наукових та інженерних завдань під час роботи з візуальними даними вимагає особливих зусиль, спираючись на знання специфічних методів. Доцільним та актуальним науково-практичним завданням є діяльність, що пов'язана з удосконаленням сучасних та розробкою нових методів цифрової обробки зображень.

Веб-дизайн та комп'ютерна графіка – актуальні для розробників штучної графіки (комп'ютерні ігри, анімація, зображення веб-сторінок) та, наприклад, розробників спеціалізованих програм для автоматизованих систем управління військами, об'єктів критичного застосування (управління атомними станціями, залізничним та авіаційним транспортом,



центрами космічних польотів). Розпізнавання образів – актуально у військовій справі, коли правильна ідентифікація військового об’єкта може кардинально змінити хід бою чи локальної операції, знаходить своє місце в археології та архівній справі, допомагає відтворити історично цінні документальні знахідки, а також при картографуванні місцевості і передачі інформації зі штучних супутників Землі.

Розробка способу комплексної оцінки та покращення якості зображень з використанням нейронних мереж обумовлена тим, що кількість зображень з кожним днем росте, зображення аналізуються, оброблюються, архівуються, зберігаються і все це виконують як великі системи, наприклад, як дата-центри, для збереження, чи дата-сайнс корпорації для обробки та виведення статистичних даних, так і компактні користувацькі пристрої як смартфон чи ноутбук.

Основна спрямованість проекту – винайдення способу, що дозволить покращити та/або прискорити комплексний аналіз та покращення зображень.

### **5.1.3 Мета та завдання проекту відповідно до проблеми.**

Метою проекту є винайдення та тестування системи для прискорення роботи кластерних систем з динамічним розподілом задач. Для цього необхідно, виділити вже існуючі методи підвищення ефективності виявити їх недоліки та переваги один над одним обрати шлях розвитку проекту відповідно до виявлених переваг та недоліків. Прискорити роботу дата-сайнс центрів та дата-центр центрів, прискорити обробку інформації, що дозволить надати проекту з винайдення методу подальшу його оптимізацію. Також ціль створити симулятор для отримання статистичної інформації для демонстрування та порівняння з аналогами винайденного методу. Відповідно

презентувати отриманні результати та знаходження інвестицій для встановлення або переустановлення подібних систем чи заміна старіших і менш оптимальних. Залучити сторонніх розробників методом подальшого випуску вільного пакету програмних засобів, що дозволять розробникам створювати власні методи чи ПЗ на основі розробленого методу, що дасть можливість використовувати та популізувати метод серед розробників.

В основу методу буду покладено засіб заміни «щоденика» та виключення статичного розпаралелювання. «Щоденик» системи буде замінено на граф на вузлах якого знаходяться операції системи а ребра графу це послідовність виконання, тобто на початку робити системи ми маємо лише набір вхідних вузлів так званих стартерів, а далі при виконання кожного вузла генерується наступний до якого по графу ребру переходить «поточний стан» і відбувається виконання вже наступного вузла з генерацією наступного.

#### 5.1.4 Дерево проблем.

Дерево проблем проекту зображено на рис. 5.1.

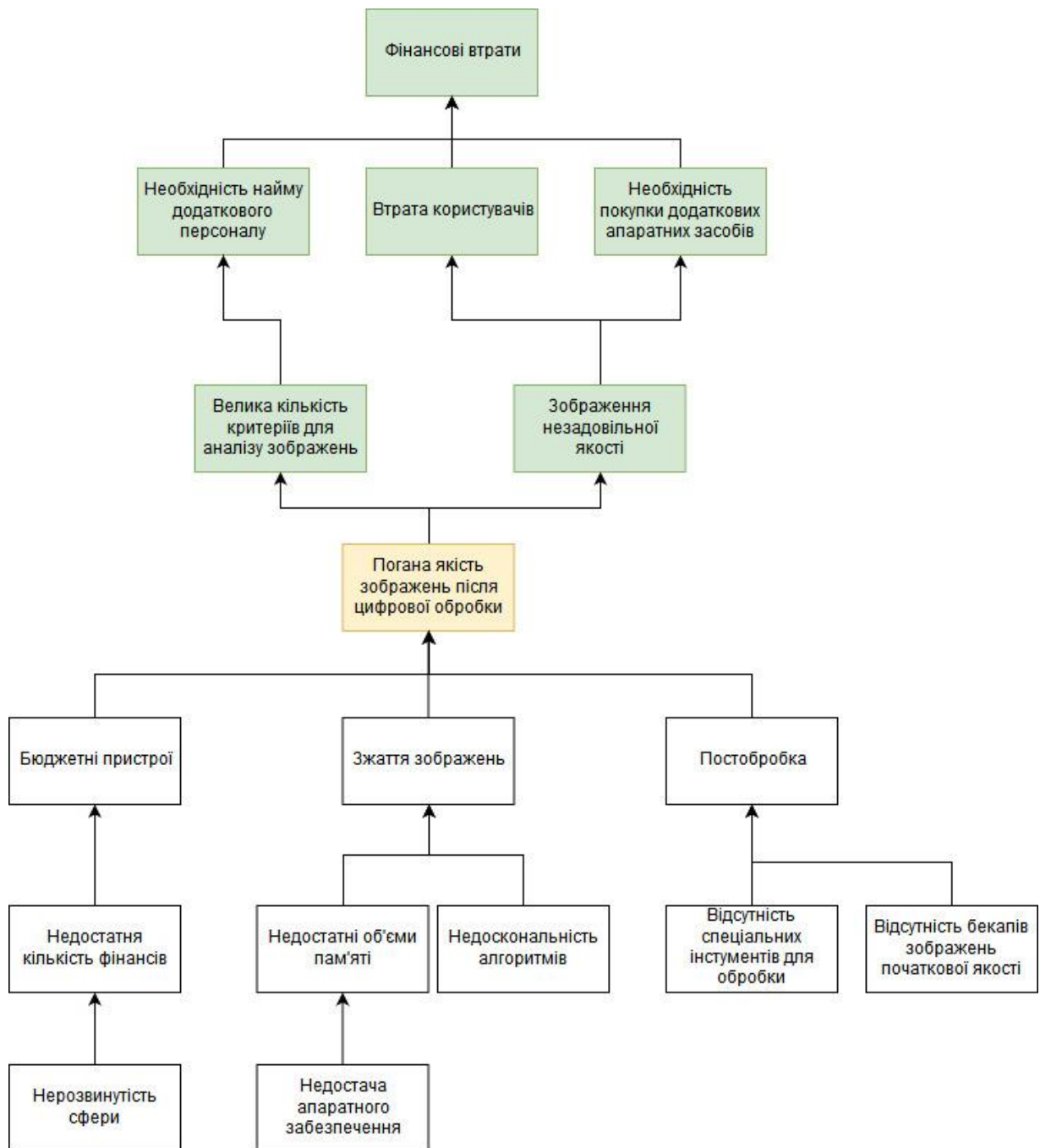


Рис 5.1 – Дерево проблем проекту.

## 5.2 Аналіз зацікавлених сторін проекту.

### 5.2.1 Зацікавлені сторони проекту

Зацікавлені сторони проекту та оцінка їх важливості та зацікавленості зібрані у таблиці 5.1.

Таблиця 5.1 – Зацікавлені сторони

Група зацікавлених осіб	Інтереси групи в проекті	Умови довгого співробітництва з проектом	Важливість	Зацікавленість
Внутрішні зацікавлені сторони проекту				
Розробник проекту	Виконання проекту; Досягнення цільових показників проекту	Подальший розвиток проекту; Нові інвестиції	10	8
Команда керування проектом	Досягнення цільових показників проекту; Подальший розвиток технологій проекту	Подальший розвиток проекту; Нові інвестиції; Збільшення особистого доходу	10	10

Таблиця 5.1 – Продовження

Інвестори проекту	Отримання зазначених доходів від участі в проекті; Подальший розвиток проекту; Досягнення цілей	Збільшення прибутку від інвестицій; Вигідніші умови підтримки проекту	10	9
Внутрішньо – корпоративні зацікавлені сторони проекту				
Менеджмент проекту	Розвиток проекту; Збереження притоку робочих місць;	Приріст робочих місць; Можливість кар'єрного росту; Реклама	7	9
Акціонери	Приріст доходності; Ріст загальної вартості проекту	Збільшення вартості проекту	10	8

Таблиця 5.1 – Продовження

Побічні співробітники	Кар'єрний ріст	Кар'єрний ріст	8	8
Зовнішні зацікавлені сторони проекту				
Дата центри	Використання продукту проекту для прискорення роботи центру; Якість власного продукту; Власний технологічний розвиток	Розвиток технологічного фону	4	6
Соціальні мережі	Використання продукту проекту для покращення якості власного продукту;	Розвиток проекту	6	9

Таблиця 5.1 – Продовження

Інші розробники ПЗ	Використання продукту для покращення власного технологічного фону	Покращення старих технологій методами внесення змін у проект	7	8
Побічні зацікавлені сторони проекту				
Розробники ПЗ	Використання технологій нейронних мереж	Оновлення технологій	8	6

### 5.2.2 Аналіз зацікавлених сторін проекту

Як видно з таблиці 5.2.1 розділу 5.2.1, зацікавлені сторони були розділені на 4 основні групи.

Внутрішні зацікавлені сторони проекту.

Тут зібрані особи, що напряду зацікавлені у проекті та його подальшому розвитку та встановленню на підприємствах закладах тощо, серед яких інвестори, для яких важливо мати прибуток з проекту, розробник проекту, який безпосередньо є керівником усього, що відбувається, як всередині команди так і зовні, команда проекту, що слідкує за виконанням проекту.

Внутрішньо-корпоративні зацікавлені сторони проекту

Ця група має менш серйозне відношення до проекту, проте без них його існування та подальший розвиток неможливі, серед них майбутні власники акцій, до яких входять і особи з першої групи, тому їх важливість та зацікавленість тісно перетинається з учасниками першої групи, а також до акціонерів належать побічні особи, зацікавленість яких може бути менша.

Побічні співробітники, їх важливість не дуже висока особливо на перших етапах розвитку, по перше їх кількість невелика і по друге вони не несуть такої сильної важливості адже не пов'язані напряму з запуском чи застосунком розроблюваного способу в певних випадках, проте якщо проект отримає подальший розвиток, їх кількість зросте, а співробітники, що були присутні на старті отримують більші доходи.

Зовнішні зацікавлені особи.

Усі, хто зацікавлений у отриманні доступу до використання методу, тобто «цільова аудиторія» методу, до неї входять усі фірми, підприємства, корпорації, заклади, що працюють з інформацією у тому, чи іншому вигляді, а також можуть бути носіями статистичних даних, або тестовими платформами методу.

Побічні зацікавлені сторони.

У супереч тому, що дана група має низьку зацікавленість та невисоку важливість, є потужним рушієм для популяризації методу, адже незалежні програмісти, можуть використовувати метод для створення власних продуктів у майбутньому, що принесе його у постійне використання.



### **5.3 Опис наукового проекту та технології.**

Науковим продуктом у рамках наукової дисертації є спосіб оцінки та покращення якості зображень за допомогою нейронних мереж, що і є науковою назвою продукту.

Зазвичай зображення, сформовані різними інформаційними системами, спотворюються дією завад. Це ускладнює як їхній візуальний аналіз, так і автоматичну обробку. При вирішенні деяких завдань обробки зображень у ролі завад можуть виступати ті або інші компоненти самого зображення. Наприклад, при аналізі космічного знімка земної поверхні може стояти завдання визначення границь між її окремими ділянками - лісом і полем, водою й сушею тощо. З погляду цього завдання окремі деталі зображення всередині розділених областей є завадою.

Ослаблення дії завад досягається фільтрацією. При фільтрації яскравість (сигнал) кожної точки вихідного зображення, спотвореного завадою, замінюється деяким іншим значенням яскравості, яке в меншій мірі було спотворене завадою. Фільтрація зображень здійснюється в просторовій і частотній областях.

При просторовій фільтрації зображень перетворення виконується безпосередньо над значеннями зображення. Результатом фільтрації є оцінка корисного сигналу зображення. Це досягається завдяки тому, зображення часто являє собою двовимірну функцію просторових координат, що змінюється по цих координатах повільніше, ніж завада, що також є двовимірною функцією. Це дозволяє при оцінці корисного сигналу в кожній точці зображення взяти до уваги сусідні точки, скориставшись певною подібністю сигналу. В інших випадках, навпаки, ознакою корисного сигналу є різкі перепади яскравості.

9	8	7	6	7	8	9
8	5	4	3	4	5	8
7	4	2	1	2	4	7
6	3	1	0	1	3	6
7	4	2	1	2	4	7
8	5	4	3	4	5	8
9	8	7	6	7	8	9

Рисунок 5.2 – Конфігурації околиці елемента "0" у кадрі зображення в ієрархічній послідовності.

Однак, як правило, частота цих перепадів відносно невелика, так що на значних проміжках сигнал або постійний, або змінюється повільно. І в цьому випадку властивості сигналу проявляються при спостереженні не тільки його окремої точки, але й при аналізі її околиці. Поняття околиці є досить умовним. На рис. 5.2 представлена ієрархія околиць відліку, позначеного "0".

"1" позначена околиця першого порядку, для якої відстань між елементами дорівнює 1. "2" позначена околиця другого порядку, до якої ставляться діагональні елементи, відстань від яких до центрального відліку "0" дорівнює  $\sqrt{2}$ . Околиця третього порядку представлена елементами, що знаходяться від центрального елемента на відстані 2, і так далі.

## 5.4 Бізнес рішення та основні характеристики бізнес - продукту.

### 5.4.1 Резюме продукту

Результатом науково – дослідницької роботи буде ПЗ та документація до нього, дане ПО допоможе покращити роботу з зображеннями, або

прискорити її. Далі буде розглянуто принцип розробки, користь продукту, способи підтримки та подальший розвиток, що дасть абстрактне розуміння про продукт, що буде отриманий в результаті дослідницької роботи. Також далі будуть розглянуті можливості та шляхи вводу ПО до систем різних типів, а також можливості отримання ПО та подальшого використання у «домашніх умовах», також будуть описані необхідні вимоги для встановлення ПЗ.

#### **5.4.2 Опис продукту**

Результатом дослідницької роботи буде ПЗ, але слід зазначити, що ПЗ далеко не основний виробничий продукт, в основі ПЗ лежить спосіб комплексної оцінки та покращення якості зображень, який можливо інтегрувати в інші рішення.

Основним плюсом при використанні способу є прискорення або покращення аналізу та обробки інформації, такий спосіб може знадобитися: дата центрам, соціальним мережам, додаткам, що працюють з фото.

Дата-центри зможуть прискорити або покращити процес обробки зображень.

Соціальні мережі зможуть запропонувати користувачам оброблені зображення.

Розробники ПЗ зможуть інтегрувати даний спосіб в додатки пов'язані з камерою або галереєю.

Основною особливістю розробки є простота в використанні та інтеграції в інші рішення спосіб буде реалізовано у вигляді додатку або бібліотеки, що підключається до іншого.

Процес «продажу» способу буде відбуватися методом інтегрування способу в інші проекти або для хмарного аналізу та обробки зображень, та подальша його підтримка за оплату послуг по підтримці, щомісяця та за використання технології, у випадку з розробниками власних проектів їм буде надана можливість скористуватися демонстрацією способу що не вийде застосувати для великих наборів даних. Для отримання доступу до технології необхідно замовити інтегрування в проект та обов'язково подальшу підтримку, до якої буде прив'язаний окремий спеціаліст.

В майбутньому спосіб перетворить в серверний додаток, який зможе аналізувати та обробляти зображення по різним критеріям.

#### **5.4.3 Конкурентні переваги**

Дане рішення є інноваційним по причині того, що і для аналізу, і для обробки зображень будуть використані нейронні мережі з розробленими алгоритмами. На даний момент існують рішення з використанням нейронних мереж для виконання специфічних завдань таких як часткове пошкодження зображення, або розмитість, але не існує засобу який би комплексно оброблював зображення.

Також нейронні мережі зі специфікою передбачення не є новими технологіями, їх дослідження почалися ще в минулому столітті, де мали значні досягнення, але на даний момент їх розвиток сповільнився.

На основі наявних досягнень цих мереж також побудовані певні програми, але всі відомі з них, не мають жодного наукового контексту, а їх ефективність жодним чином не доведена.

Серед основних критеріїв у конкурентної спроможності є:

1. Патентна здатність (новизна проекту).

2. Раціональність виробничої організаційної структури схеми.
3. Конкурентний персонал.
4. Прогресивність технології.
5. Прогресивність технологічних процесів та обладнання.
6. Науковий рівень розвитку розробників.
7. Науковий рівень системи.
8. Вимогливість до апаратного забезпечення.
9. Швидкодія.

Технологічні переваги способу(проекту):

1. Наявні засоби для комплексної оцінки великої кількості зображень за допомогою нейронних мереж.
2. Позбавлення від проблеми лінійного алгоритмічного аналізу, який не реагує на виняткові ситуації, тобто програміст не повинен модифікувати код програми після того як вона зустріли новий нетиповий випадок.
3. Реалізація у вигляді додатку та серверного застосунку надасть можливість розповсюджувати рішення на мобільні пристрої, дата-центри, та соціальні мережі.
4. Зручний пакет розробника, що надасть додаткову мотивацію до використання методу «вільним» та побічним розробникам.
5. Кроссплатформність, даний метод, а саме його програмна реалізація не потребує підключення до мережі інтернет, тобто працює без підключення до мережі, навіть локальної.
6. Відповідно по пункту 5, програмна реалізація методу має збільшену відмовостійкість, при відсутності з'єднання з мережею Інтернет відключаються лише побічні функції ПЗ, а саме покращення зображень залишається доступним.

Загальні переваги проекту:

1. Так як проект та в майбутньому команда(фірма, компанія, тощо) не використовує підхід кредитування тому може дозволити зручний та сталий формат формування цінової політики.
2. Весь персонал, що буде займатися обслуговуванням та оновленням систем та обладнання у замовників, проходять постійні курси підготовки та розвитку, прийматимуть участь у конференціях.
3. Оптимізація потужностей, через те що спосіб буде покращуватись та знаходиться у постійному процесі збору даних за допомогою нейронних мереж, надасть можливість оновлювати ПО та обладнання у найкоротші строки, а також підвищувати якість послуг без участі розробників, або за допомогою мінімального вплив останніх.
4. Відсутність некваліфікованих співробітників.
5. Встановлення та оновлення системи відбувається на користувацьких пристроях напряду без посередників, а на серверах за допомогою декількох макросів;
6. Онлайн тестування методу на прикладах.

Також слід приділити особливу увагу пункту про тестування, адже при впровадженні подібних методів на виробництво, замовник хоче бути впевненим у тому, що дана система буде давати необхідні результати, це проблема вирішується методом реалізації окремої повноцінної, самостійної користувацької бібліотеки тестування, яка надасть змогу окрім симуляції роботи системи на реальних даних та моніторингу, побудувати систему покращення роботи підприємства, а саме програмного забезпечення на яке вона встановлюється. Даний аспект надає суттєву перевагу, адже перед встановленням метод може бути повністю протестований, а результати покращення проаналізовані замовником.

#### **5.4.4 Клієнти. Сегменти ринку споживання**

Наших майбутніх клієнтів можна сегментувати та виділити по наступним критеріям:

1. Географічний – припускає, що люди, які проживають в різних країнах, регіонах, містах, районах мають різні запити, реакції на продукт в зв'язку з відмінними кліматичними умовами, способом життя, культурними особливостями. Не часто застосовується самостійно;
2. Соціально-демографічний – розділяє людей виходячи з таких показників, як стать, вік, сімейний стан, освіту, рівень доходу, рід заняття, які мають безпосередній вплив на формування споживчих запитів. Найбільш поширений і часто використовуваний метод;
3. Психографічний – об'єднує клієнтів в групи на підставі інтересів, цінностей, стилю життя. Доповнює географічну, соціально-демографічну сегментацію і дозволяє повніше розкрити потреби покупців. У той же час вважається самим важко вимірюваним видом, так як описує внутрішні переживання, емоції людини;
4. Поведінковий – базується на визначенні сегментів виходячи з мотивації покупки, лояльності бренду, готовності до замовлення продукту. Найбільш ефективний метод сегментації, так як піддається виміру. Застосовується в зв'язці з іншими видами.

У свою чергу, поведінковий тип сегментування умовно можна розділити на такі підвиди:

1. За обставинами застосування - припускає розподіл клієнтів на частини відповідно до виникненням ідеї здійснення покупки, застосування продукту.

2. На базі вигоди – поділ клієнтів виходячи з користі, вигоди, яку споживач шукає в товар / послугу.
3. На основі інтенсивності використання - має на увазі розподіл на групи активних, помірних, слабких покупців.
4. З урахуванням ступеня лояльності – визначає рівень прихильності клієнтів до бренду, марці продукту.
5. За статусом клієнта – розподіл користувачів продукту на групи колишніх, потенційних, новачків, регулярних і т.д .
6. По стадії готовності споживача - класифікує клієнтів на необізнаних, обізнаних, незацікавлених, зацікавлених.

Виходячи з вищезазначеної інформації, ми можемо сегментувати наших клієнтів наступним чином:

#### **5.4.4.1 Географічно**

Ми не маємо певного напрямленого ринку чи споживачів. Програмний додаток буде розроблений та працюватиме повністю на англійській мові, яка є міжнародною для будь-якого користувача або сервісу пов'язаного зі сферою медіа контенту, до яких належить ідея та наявна реалізація аналізу та обробки зображень.

Також, після успішного старту – програма буде локалізована для всіх країн, де будуть наявні споживачі, так як мовний контент програми незначний та легко перекладається.

#### **5.4.4.2 Соціально-демографічно**

Програмний метод буде спрямовано в першу чергу на молоду та середньовікову групу, тобто основна група користувачів буде мати вік від 12 до 40 років, з яких буде явно домінувати молодша група віком від 12 до 25 років.



По статі програма більш орієнтована на жіночу стать в користувацькому сегменті, так як статистично жінки щоденно роблять більше фото ніж чоловіки.

Освіта та рід занять клієнта також має основне напрямлення – обробка зображень та фото.

А рівень доходу майбутнього споживача буде менше середнього у відношенні до середнього доходу в країні, де він проживає. Так як саме в такій ситуації люди шукають способи додаткової обробки зображень, окрім засобів що надаю наявний пристрій з камерою.

#### **5.4.4.3 Психографічно**

По даному критерію неможливо дати точних оцінок майбутніх клієнтів, попри те, що вже було сказано вище: життя людини та його інтереси у домінуючому випадку будуть так чи інакше пов'язані зі сферою медіа та активного використання соціальних мереж.

#### **5.4.4.4 Поведінково**

У поведінкових шаблонах наших можливих клієнтів не можна виділити щось зверх того, що було описано по попереднім критеріям.

У результаті сегментації ринку споживачів ми можемо сказати, що середньостатистичним користувачем буде людина віком від 12 до 25 років, яка активно накопичує фотоматеріали та зображення, або ж має велику кількість наведених вище матеріалів.

Тож наш початковий програмний продукт буде мати доволі конкретні інтерфейс, дані, модель роботи та платформу розповсюдження, зважаючи на підготовленість до них користувача.

#### **5.4.5 Унікальна ціннісна пропозиція**

Щоб відкрити сутність нашої ціннісної пропозиції слід відповісти на два головних питання:

1. Які можливості надає наш продукт клієнту?
2. Чому йому варто обрати саме наш продукт?

Отже спершу слід підсумувати, яку саме проблему клієнта наш програмний вирішить, та чого за допомогою нього від зможе досягти.

Виходячи з усього опису та мети майбутньої системи, можна перелічити наступне:

1. Ми збираємо дані з усіх користувачів продукту, тобто всі зображення, які потрапляють до сервісу навчають нейронну мережу, що в майбутньому покращую якість вихідних зображень;
2. и опрацьовуємо їх спеціально розробленим алгоритмом за технологією BigData для виділення, класифікації та різноманітного групування ключових сегментів інформації для отримання найбільш інформативної вибірки;
3. Готова до роботи нейронна мережа міститься в розповсюджуваному продукті, а її клони й надалі проходять навчання на вибірках, які вже включають у себе прогнози програми реліз-кандидата та їх результати для подальшого оновлення та покращення мережі у майбутніх версіях продукту;
4. Наш продукт також збирає аналітику про роботу клієнта, його успіхи та задоволеність, та надає повноцінний зворотній зв'язок, для покращення програми та виправлення помилок. А також публікування відгуків у соціальних мережах для більшого розголосу та відкритості;

5. Як результат – наш продукт надає кінцевим користувачам якісні дані, а сервісам додаткові можливості.

Чому користувачу слід обрати саме наш продукт?

По-перше – тому що на даний момент у нашої системи попросту немає конкурентів. А якщо брати до уваги існуючі аналоги з не підтвердженою ефективністю, до яких навіть немає відкритого доступу. Також слід зазначити, що навіть клієнтські рішення не завжди безкоштовні, а можливості перевірити їх до покупки немає.

### **5.5 Доходи та витрати**

Звичайно будь-який реально працюючий додаток, який має під собою спеціально розроблені алгоритми, та має на меті таку амбітну ціль, як прогнозування курсу акцій, або ж професійна робота з графікою, буде мати витрати на його розробку, патентування, втілення власного дизайну, маркетингу, розповсюдження та підтримки, як в технічному плані, так і в сенсі зворотного зв'язку з клієнтами.

Наш програмний продукт не є виключенням, його ідея виникла не на порожньому місці, а перші напрацювання та макет майбутньої системи потребували досить багато часу, та наявності чималого досвіду у технологіях нейронних мереж, основах економіки, концепції поняття машинного навчання. До цього було проаналізовано безліч існуючих інформаційних ресурсів, на предмет думок спільноти.

Лише після цього був сформований план дій, та каркас майбутньої системи, модель її роботи, чинники, які допоможуть досягти ефективності та приблизні рамки функціональності системи.

Але це тільки початок, попереду найважливіше – етап розробки продукту, не тільки програмного, але й наукового.

Для цього звичайно краще найняти спеціальних людей, які зможуть зробити це ефективним.

Та звичайно для завершення продукту, та його повноцінної підтримки знадобиться допомога інших розробників, тестувальників та працівників технічної підтримки, коли додаток буде розвиватися у повноцінний сервіс.

Все це приводить нас до чималих витрат, тож ми маємо скласти план їх покриття.

До моменту альфа-версії продукту – його винахідник та головний розробник бере всі обов'язки по розробці на себе, щоб мати в результаті мінімально працюючий продукт, який згодом буде виставлений на сайті [www.kickstarter.com](http://www.kickstarter.com) для пошуку інвесторів.

Весь процес розробки буде вестися з підтримкою засобів контролю версій, а відкритий код програми буде викладатися на сервери [www.github.com](http://www.github.com).

Після чого, система буде представлена вченій раді магістерського захисту у технічному вищому навчальному закладі України – НТУУ «КПІ імені Ігоря Сікорського» для підтвердження наукової цінності продукту та його готовності для реального світу.

Базова версія програми вже буде мати необхідну для роботи функціональність, тож вона буде виставлена у якості безкоштовного додатку у відкритий доступ через магазин додатків Microsoft, одразу ж після її валідації, згідно їх умовам.

На цьому етапі ми маємо досягнуту першу мету і надалі все залежить від успіху у зборах коштів на подальшу роботу.

Якщо по певним причинам, їх не вдасться зібрати, продукт надалі буде підтримуватися головним розробником, але більше в якості персональної програми та професіонального інтересу, а його публічний розвиток буде завершено.

У випадку успішного фінансування зі сторони спільноти – на перші 40% бюджету буде розпочата рекламна компанія, а по мірі надходження решти коштів – будуть найматися решта команди у кількості ще двох спеціалізованих розробників, та одного тестувальника.

Отже основні витрати – це оплата праці додаткових працівників у кількості 3-4 чоловік на другому етапі проекту, а також рекламної компанії, які попередньо оцінюються в 1500 у. о.

Після чого, третій етап передбачає виставлення фінальної версії продукту вже на основі платної моделі в основі якої буде надання доступу до програми на певний оплачуваний період.

### **5.5.1 Доходи**

Як зрозуміло з попередніх пунктів та ринку споживачів, товаром є метод для підвищення ефективності роботи паралельних кластерних систем з динамічним розподілом задач. Основними джерелом доходу будуть кошти з експлуатації даного методу. Загальну дохідність проекту в майбутньому буде прийнято рахувати метриками доходності, такими, як:

1. Bookings (контракти).
2. Revenue (дохід).
3. TCV – Total Contract Value (загальна вартість контракту).

4. LTV – Life Time Value (тотальна цінність).
5. Unearned or Deferred Revenue (майбутні доходи).

Bookings – це оцінка вартості контракту між компанією і клієнтом. Ця метрика відбиває зобов'язання клієнта заплатити компанії зазначені в договорі гроші. Так як основою експлуатації та впровадження на виробництво методу є контракт дана метрика, грає основну роль в обрахунках доходності проекту, сумарна вартість контракту на один заклад.

Про прибуток (Revenue) можна говорити тоді, коли послуга вже надана або буде надаватися регулярно протягом зазначеного в договорі терміну підписки. Є вихідним фактором першої метрики, рахується по закінченню контракту, тобто при вдалому завершенні.

TCV – вихідна вартість контракту, рахується у випадку, якщо у контракті описані пункти підтримки, доповнення або перерахування вартості чогось, наприклад оплата подальшої підтримки виробництва.

TCV може з часом збільшуватися або зменшуватися. Переконайтеся, що TCV враховує також одноразові витрати, оплату спеціальних послуг і повторювані платежі.

LTV – довічна цінність - це поточна оцінка майбутньої чистого прибутку від клієнта протягом усього періоду його відносин з компанією. Вона допомагає визначити довгострокову цінність клієнта, а також чистий прибуток в розрахунку на одного клієнта з урахуванням витрат на його залучення (CAC). Місячна маржа від кожного клієнта = дохід від клієнта мінус змінні витрати, пов'язані з клієнтом. Змінні витрати включають в себе всі адміністративні та операційні витрати, пов'язані з обслуговуванням клієнта.

### **5.5.2 Витрати**

Для прийняття рішення щодо інвестиційного проекту всі витрати, пов'язані з його здійсненням, необхідно розподілити на інвестиційні та виробничі.

Загальна сума витрат на здійснення проекту включає:

1. Витрати на формування основного капіталу містять початкові і поточні інвестиції.
2. Витрати на формування оборотного капіталу.
3. Виробничі витрати.

Всі інвестиційні потреби підприємства можна поділити на три групи:

1. Прямі інвестиції – безпосередньо необхідні для реалізації інвестиційного проекту.
2. Супутні інвестиції – вкладення в об'єкти, безпосередньо технологічно які пов'язані із забезпеченням нормальної експлуатації.
3. Інвестування виконання науково-дослідних робіт.

До складу початкових інвестицій відносяться:

1. Витрати на передінвестиційні дослідження, проведення дослідницьких і конструкторських робіт, на розробку проектних матеріалів, на робоче проектування і прив'язку проекту.
2. Витрати на придбання та оренду потужностей, включаючи вартість підготовки до освоєння.
3. Витрати на придбання та доставку машин, обладнання, інструменту та інвентарю, в тому числі імпортних.
4. Витрати на приймально-здавальні випробування.

5. Витрати на пусконаладжувальні роботи, комплексне освоєння проектних потужностей і досягнення проектних техніко-економічних показників.
6. Витрати на придбання патентів, ліцензій, ноу-хау, технологій та інших амортизаційних нематеріальних активів.
7. Витрати на підготовку кадрів для об'єктів, що вводяться в дію.
8. Одноразові виплати, зокрема гарантує і страховим організаціям.
9. Витрати, що виникають при створенні та реєстрації фірми (оплата юридичних послуг зі складання установчих документів, витрати на реєстрацію фірми і оформлення прав власності).
10. Витрати на підготовчі дослідження не враховано в кошторисної вартості об'єкта.
11. Витрати, пов'язані з діяльністю персоналу в період підготовки виробництва (оплата праці, витрати, утримання приміщень, автомобілів, комп'ютерів та іншого обладнання), не враховані в кошторисної вартості об'єкт.

Висновки по витратам:

Вартість проекту = ціна проектування + ціна розробки + активна підтримка і доопрацювання після впровадження + плата за компоненти та сервіси + реклама і просування.

### **5.5.3 Таблиця з витратами / доходами**

Таблиця 5.2 описує етапи розвитку проекту, розділи витрат та сумарний дохід. Слід врахувати, що перший та другий етапи є разовими і є при проектними, третій етап розраховується помісячно, після закінчення перших двох етапів.



Таблиця 5.2 – етапи розвитку проекту

Етап проекту	Статті витрат	Шляхи доходу	Витрати	Дохід	Сума
Етап розробки	Розробка, оплата сервісів, патентування, маркетинг		1.000 у.о.	0	1.000у.о
Етап впровадження	Маркетинг, підтримка, реалізація, впровадження	Впровадженн я у тестовому режимі на сторонні, Play Market.	2.000 у.о.	6.000 у.о.	4.000 у.о.
Етап експлуатації	Підтримка, обслуговуванн я, розвиток проекту, оплата персоналу	Замовлення, о підтримка, підписки, Play Market.	2.000 у.о	15.00 0 у.о	+13.000 у.о

## 5.6 Бізнес-модель.

На рисунку (Рис. 5.3) зображена загальна схема бізнес моделі.

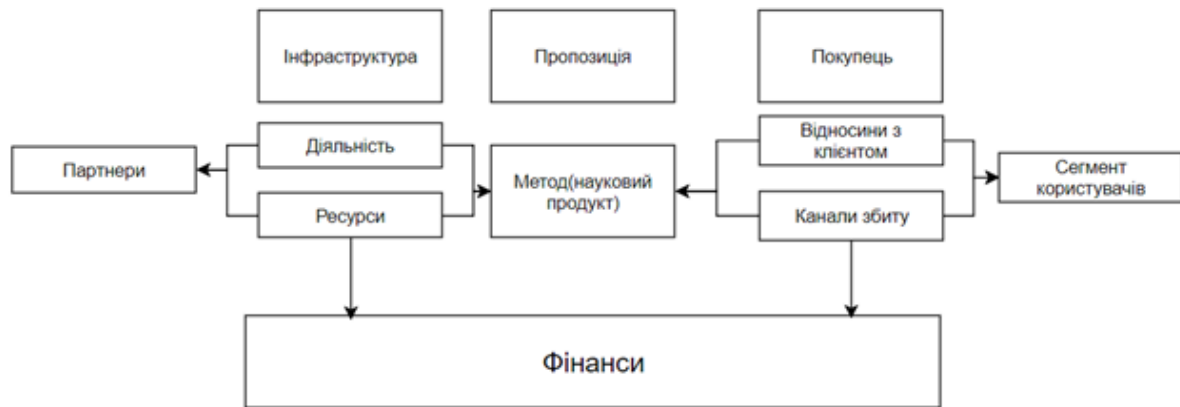


Рис. 5.3 Загальна бізнес модель.

Сегмент користувачів:

Для кого: Заклади з великою кількістю медіа даних.

Найважливіші клієнти: Великі заклади та корпорації з укладанням контракту на подальшу підтримку та оновлення.

Пропозиція:

Яку цінність від методу отримують клієнти: краща якість медіа контенту, а саме зображень на серверах. Для користувачів - обробка власних фото.

Яку проблему користувачів ми вирішуємо: неякісні фотографії/фотографії незадовільної якості.

Яку необхідність користувачів ми задовольняємо: Користувачам необхідно оновлювати, зберігати, оброблювати зображення, метод дозволяє це зробити.

Канали розповсюдження:

Через які канали ми хочемо досягти кожен сегмент наших споживачів: Google Play, реклама та звернення до менеджерів соціальних мереж.

Який з каналів розповсюдження є найпривабливішим? – можливість розповсюдження застосунку через менеджери додатків Android.

Потоки доходу:

За що готові платити користувачі: за комплексний аналіз зображень та їх покращення.

Яким чином вони можуть платити: так, як основними клієнтами є заклади, а не цільові одиночні клієнти, то більшість оплата буде укладатися за контрактами в яких буде описано умови підтримки, тестування та оновлення, контракт є можливість продовжувати, або заключати, як підписку на певний статичний час.

Альтернативні шляхи: дохід також може бути з додатку для Android.

Основні ресурси:

Серед необхідних ресурсів, платформи для тестування, та статистичні дані закладів по обробці даних. Наукові досягнення та патенти на подібні методи та фінансові ресурси.

Шляхи знаходження необхідних ресурсів: безкоштовні тестові пакети для закладів, від чого виграє проект і заклади.

Відносини з клієнтами: замовник підписуючи один з видів контракту, отримує набір послуг.

Потоки доходів: основний дохід буде надходити від сервісів які мають велику кількість зображень, які потребують покращення. Компанії оформлюють підписку у вигляді контракту за деяким пакетом послуг, які коштують по різному і містять різні набори послуг, або усі разом.

### **5.7 Висновок до п'ятого розділу**

На даний момент існує безліч загальновідомих алгоритмів та засобів для фільтрації зображень, але ті, які побудовані на нейронних мережах переважно не є публічними і або потребують навчання на великих наборах даних. Для введення технології в бізнес процеси достатньо одного спеціаліста. Оскільки даний спосіб відносно багатьох інших більш швидкодійний, його доцільно буде примінити в багатьох процесах. Тому дослідження і розробка є доцільними.

## **ВИСНОВОК**

В межах магістерської дисертації було проведено аналіз способів вирішення задачі підвищення якості зображень, а також метрик оцінки якості зображень. Він показав, що незважаючи на факт існування швидких і якісних алгоритмів, з'являються нові методи та способи які показують об'єктивно кращі результати. Було запропоновано комплексну оцінку якості оброблених зображень відносно оригіналу та способів підвищення якості зображень. А також розроблено програмне забезпечення для оцінки та покращення якості зображень.

Подальше дослідження має бути зосереджене на пошуку програмних оптимізацій для реалізації запропонованого способу. Доцільно розглянути можливості застосування більш складних фільтрів, провести експерименти з додатковими аргументами нейронної мережі, включити навчання.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Schmidhuber J. Deep learning in neural networks: An overview [Text] //Neural networks. – 2015. – Т. 61. – С. 85-117.
2. Sietsma J., Dow R. J. F. Creating artificial neural networks that generalize [Text] // Neural networks. – 1991. – Т. 4. – №. 1. – С. 67-79.
3. Cover T., Hart P. Nearest neighbor pattern classification [Text] // IEEE transactions on information theory. – 1967. – Т. 13. – №. 1. – С. 21-27.
4. R. Keys. Cubic convolution interpolation for digital image processing [Text] // IEEE Transactions on Acoustics, Speech, and Signal Processing. 29 (6): 1153—1160.
5. Hwang J. W., Lee H. S. Adaptive image interpolation based on local gradient features [Text] //IEEE Signal Processing Letters. – 2004. – Т. 11. – №. 3. – С. 359-362.
6. Dong C. et al. Image super-resolution using deep convolutional networks [Text] //IEEE transactions on pattern analysis and machine intelligence. – 2016. – Т. 38. – №. 2. – С. 295-307.
7. Romano Y., Isidoro J., Milanfar P. RAISR: rapid and accurate image super resolution [Text] // IEEE Transactions on Computational Imaging. – 2017. – Т. 3. – №. 1. – С. 110-125.
8. The Most Popular Language For Machine Learning and Data Science Is ... [Online]. Available : <http://www.kdnuggets.com/2017/01/most-popular-language-machine-learning-data-science.html>
9. Бібліотека глибокого навчання Tensorflow. [Online] Available : <https://habrahabr.ru/company/ods/blog/324898/>
10. LeCun Y. LeNet-5, convolutional neural networks [Text] / LeCun Y. // Retrieved June. – 2016. – Т. 1.

11. Ledig C. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network [Text] / C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi. // CVPR. – 2017. – T. 2. – № 3
12. Dong C. Image Super-Resolution Using Deep Convolutional Networks [Text] / C. Dong, C. Change Loy, K. He, X. Tang. // IEEE transactions on pattern analysis and machine intelligence. – 2016. – T. 38. – № 2. – pp. 295-307.
13. Dong C. Learning a deep convolutional network for image super-resolution [Text] / Dong C. // Proc. ECCV. – 2014. – p. 184-199.
14. Zhang C. et al. Understanding deep learning requires rethinking generalization [Text] //arXiv preprint arXiv:1611.03530. – 2016.
15. Sunner Li Simple Introduction about Hourglass-like Model [Text] / Li Sunner // Available : <https://medium.com/@sunnerli/simple-introduction-about-hourglass-like-model-11ee7c30138>, 2017.
16. Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift [Text] //arXiv preprint arXiv:1502.03167. – 2015.
17. Xu B. et al. Empirical evaluation of rectified activations in convolutional network [Text] // arXiv preprint arXiv:1505.00853. – 2015.
18. Bevilacqua M. et al. Low-complexity single-image super-resolution based on nonnegative neighbor embedding [Text] . – 2012.
19. Ronneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation [Text] //International Conference on Medical image computing and computer-assisted intervention. – Springer, Cham, 2015. – C. 234-241.

20. Hore A., Ziou D. Image quality metrics: PSNR vs. SSIM //Pattern recognition (icpr), 2010 20th international conference on. – IEEE, 2010. – C. 2366-2369.
21. Wang Z. et al. Image quality assessment: from error visibility to structural similarity //IEEE transactions on image processing. – 2004. – T. 13. – №. 4. – C. 600-612.
22. Li X., Cai J. Robust transmission of JPEG2000 encoded images over packet loss channels //Multimedia and Expo, 2007 IEEE International Conference on. – IEEE, 2007. – C. 947-950.
23. Thomos N., Boulgouris N. V., Strintzis M. G. Optimized transmission of JPEG2000 streams over wireless channels //IEEE Transactions on image processing. – 2006. – T. 15. – №. 1. – C. 54-67.



## ДОДАТКИ

## **ДОДАТОК 1**

**Вихідний код стр .ру**

```

import cv2
import numpy as np from math import inf from config import *

def getMSE(image1, image2):
    s1 = cv2.absdiff(image1, image2) s1 = np.float32(s1)
    s1 = cv2.multiply(s1, s1) s = cv2.sumElems(s1)
    valSum = s[0] + s[1] + s[2]
    return valSum / (len(image1.shape) * image1.shape[0] * image1.shape[1])

def getPSNR(image1, image2):
    mse = getMSE(image1, image2) if mse != 0.0:
        pixel_max = 255.0
        return inf
    def getSSIM_one_channel(image1, image2): k1 = 0.01
    C1 = pow(k1 * (pow(2, 8) - 1), 2)
    k2 = 0.03
    C2 = pow(k2 * (pow(2, 8) - 1), 2)
    image1 = np.float32(image1) image2 = np.float32(image2)
    image1_sq = cv2.pow(image1, 2) image2_sq = cv2.pow(image2, 2)
    image1_image2 = cv2.multiply(image1, image2) kSize = (11, 11)
    sigmaX = 1.5
    mu1 = cv2.GaussianBlur(image1, kSize, sigmaX) mu2 =
    cv2.GaussianBlur(image2, kSize, sigmaX)
    mu1_sq = cv2.pow(mu1, 2) mu2_sq = cv2.pow(mu2, 2)
    mu1_mu2 = cv2.multiply(mu1, mu2)
    sigma1_sq sigma1_sq = cv2.GaussianBlur(image1_sq, kSize, sigmaX)
    = cv2.addWeighted(sigma1_sq, 1, mu1_sq, -1,
    0)
    sigma2_sq sigma2_sq = cv2.GaussianBlur(image2_sq, kSize, sigmaX)
    = cv2.addWeighted(sigma2_sq, 1, mu2_sq, -1,
    0)
    sigma12 = sigma12 = cv2.GaussianBlur(image1_image2, kSize, sigmaX)
    cv2.addWeighted(sigma12, 1, mu1_mu2, -1, 0)
    tmp1 = cv2.multiply(mu1_mu2, 2)
    tmp1 = cv2.add(tmp1, C1)
    tmp2 = cv2.multiply(sigma12, 2) tmp2 = cv2.add(tmp2, C2)
    tmp3 = cv2.multiply(tmp1, tmp2)
    tmp1 = cv2.add(mu1_sq, mu2_sq) tmp1 = cv2.add(tmp1, C1)
    tmp2 = cv2.add(sigma1_sq, sigma2_sq) tmp2 = cv2.add(tmp2, C2)
    tmp1 = cv2.multiply(tmp1, tmp2) ssim_map = cv2.divide(tmp3, tmp1) return
    cv2.mean(ssim_map)
    def getSSIM(image1, image2):
    ssim = (getSSIM_one_channel(image1[...., 0], image2[...., 0])[0]
    + getSSIM_one_channel(image1[...., 1], image2[...., 1])[0]
    + getSSIM_one_channel(image1[...., 2], image2[...., 2])[0]) / 3 return
    ssim
    def main():
    orig = os.listdir(test_folder) images = {}
    results = {}
    for name, folder in res_folders.items(): images[name] =
    os.listdir(folder)
    results[name] = {'PSNR': [], 'SSIM': [], 'MSE': []}
    for i, file in enumerate(orig):
    image = cv2.imread(os.path.join(test_folder, file), cv2.IMREAD_COLOR)
    for name, image in images.items():
    res_image = cv2.imread(os.path.join

```

```

(res_folders[name], images[name][i]), cv2.IMREAD_COLOR)
results[name]['PSNR'].append(getPSNR(image, res_image))
results[name]['SSIM'].append(getSSIM(image, res_image))
results[name]['MSE'].append(getMSE(image, res_image))
import csv
for name, values in results.items(): print(name + ':')
print('PSNR = ' + str(np.mean(values['PSNR'])))
print('MSE = ' + str(np.mean(values['MSE'])))
with open('results_{0}.csv'.format(name), 'w', newline='') as csvfile:
    filewriter = csv.writer(csvfile, delimiter=';')
    filewriter.writerow(values['PSNR']) filewriter.writerow(values['SSIM'])
    filewriter.writerow(values['MSE'])

if name == " main ": main()

```

## **ДОДАТОК 2**

**Вихідний код unet.py**

```

import torch
import torch.nn as neunet
import torch.nn.functional as F
from .common import *

class UNet(neunet.Module):

    def __init__(self, num_input_channels=3, num_output_channels=3,
                  feature_scale=4, more_layers=0, concat_x=False,
                  upsample_mode='deconv', pad='zero',
norm_layer=neunet.InstanceNorm2d, need_sigmoid=True, need_bias=True):
        super(UNet, self).__init__()

        self.feature_scale = feature_scale
        self.more_layers = more_layers
        self.concat_x = concat_x

        filt = [64, 128, 256, 512, 1024]
        filt = [x // self.feature_scale for x in filt]

        self.start = unetConv2(num_input_channels, filt[0] if not
concat_x else filt[0] - num_input_channels, norm_layer, need_bias, pad)

        self.downsamp1 = unetdownsamp(filt[0], filt[1] if not concat_x
else filt[1] - num_input_channels, norm_layer, need_bias, pad)
        self.downsamp2 = unetdownsamp(filt[1], filt[2] if not concat_x
else filt[2] - num_input_channels, norm_layer, need_bias, pad)
        self.downsamp3 = unetdownsamp(filt[2], filt[3] if not concat_x
else filt[3] - num_input_channels, norm_layer, need_bias, pad)
        self.downsamp4 = unetdownsamp(filt[3], filt[4] if not concat_x
else filt[4] - num_input_channels, norm_layer, need_bias, pad)

        if self.more_layers > 0:
            self.more_downsamps = [
                unetdownsamp(filt[4], filt[4] if not concat_x else
filt[4] - num_input_channels, norm_layer, need_bias, pad) for i in
range(self.more_layers)]
            self.more_ups = [unetUp(filt[4], upsample_mode, need_bias,
pad, same_num_filt=True) for i in range(self.more_layers)]

            self.more_downsamps = ListModule(*self.more_downsamps)
            self.more_ups = ListModule(*self.more_ups)

        self.up4 = unetUp(filt[3], upsample_mode, need_bias, pad)
        self.up3 = unetUp(filt[2], upsample_mode, need_bias, pad)
        self.up2 = unetUp(filt[1], upsample_mode, need_bias, pad)
        self.up1 = unetUp(filt[0], upsample_mode, need_bias, pad)

        self.final = conv(filt[0], num_output_channels, 1,
bias=need_bias, pad=pad)

        if need_sigmoid:
            self.final = neunet.Sequential(self.final, neunet.Sigmoid())

    def forward(self, inputs):

        # downsampsample
        downsamps = [inputs]

```



```

        self.conv2= neunet.Sequential(conv(out_size, out_size, 3,
bias=need_bias, pad=pad),
                                     norm_layer(out_size),
                                     neunet.ReLU(),)
    else:
        self.conv1= neunet.Sequential(conv(in_size, out_size, 3,
bias=need_bias, pad=pad),
                                     neunet.ReLU(),)
        self.conv2= neunet.Sequential(conv(out_size, out_size, 3,
bias=need_bias, pad=pad),
                                     neunet.ReLU(),)

    def forward(self, inputs):
        outputs= self.conv1(inputs)
        outputs= self.conv2(outputs)
        return outputs

class unetdownsamp(neunet.Module):
    def __init__(self, in_size, out_size, norm_layer, need_bias, pad):
        super(unetdownsamp, self).__init__()
        self.conv= unetConv2(in_size, out_size, norm_layer, need_bias,
pad)
        self.downsamp= neunet.MaxPool2d(2, 2)

    def forward(self, inputs):
        outputs= self.downsamp(inputs)
        outputs= self.conv(outputs)
        return outputs

class unetUp(neunet.Module):
    def __init__(self, out_size, upsample_mode, need_bias, pad,
same_num_filt=False):
        super(unetUp, self).__init__()

        num_filt = out_size if same_num_filt else out_size * 2
        if upsample_mode == 'deconv':
            self.up= neunet.ConvTranspose2d(num_filt, out_size, 4,
stride=2, padding=1)
            self.conv= unetConv2(out_size * 2, out_size, None,
need_bias, pad)
        elif upsample_mode=='bilinear' or upsample_mode=='nearest':
            self.up = neunet.Sequential(neunet.Upsample(scale_factor=2,
mode=upsample_mode),
                                     conv(num_filt, out_size, 3,
bias=need_bias, pad=pad))
            self.conv= unetConv2(out_size * 2, out_size, None,
need_bias, pad)
        else:
            assert False

    def forward(self, inputs1, inputs2):
        in1_up= self.up(inputs1)

        if (inputs2.size(2) != in1_up.size(2)) or (inputs2.size(3) !=
in1_up.size(3)):
            diff2 = (inputs2.size(2) - in1_up.size(2)) // 2
            diff3 = (inputs2.size(3) - in1_up.size(3)) // 2
            inputs2_ = inputs2[:, :, diff2 : diff2 + in1_up.size(2),

```



```
diff3 : diff3 + in1_up.size(3)]  
    else:  
        inputs2_ = inputs2  
  
    output= self.conv(torch.cat([in1_up, inputs2_], 1))  
    return output
```